MAYO 1989

REVISTA PARA USUARIOS DE LA NORMA

LOAD MSX

AÑO 3 Nº 37 ₳ 200 REP. ARGENTINA



3 UTILITARIOS

BOLETIN ELECTRONICO PARA MSX

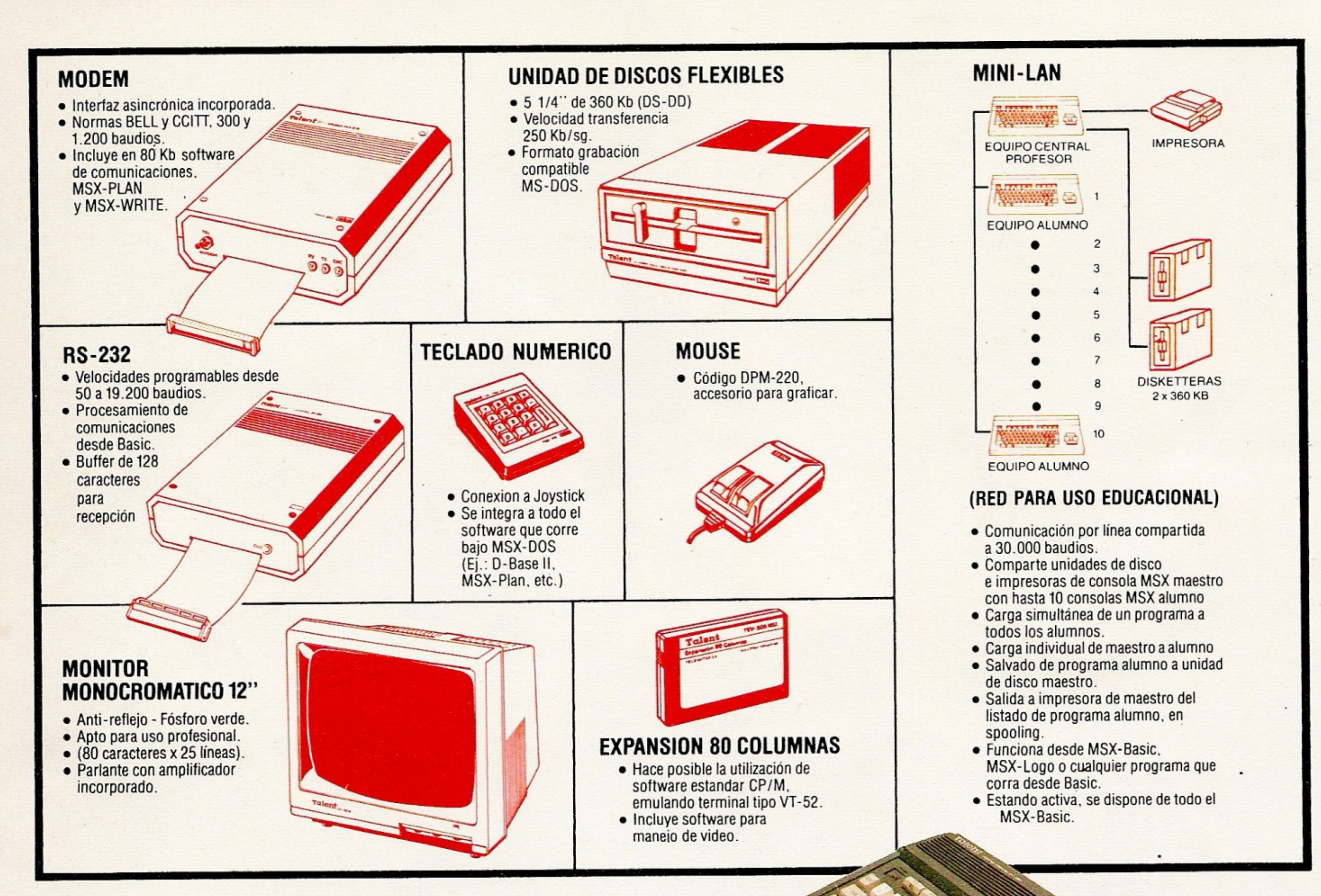
TURBO BASIC: MAS VELOCIDAD



GRAFICOS PARA MSX2

OFRECEMOS 30 BECAS DE TRABAJO

Encienda una computadora Talent MSX y sus periféricos.



Software

MSX-LOGO	MSX-LPC	MSX-PLAN	MSX-WRITE
Desarrollado por Logo Computer System Inc. con aplicación de primitivas y redacción del Manual por los Ings. Hilario Fernández Long y Horacio Reggini.	Lenguaje de programación estructurado y en castellano.	Planilla de calculo de Microsoft Corp. (Versión para MSX del Multiplan.)	Procesador de palabra de ASCII Corp. en castellano.

MSX DPC-200

Talent

Tecnología y Talento *en su casa*

Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina.
6 meses de garantía y mensualmente en su quiosco la revista Load MSX.

• **MSX, MSX-DOS, MSX-PLAN, MS-DOS**, son marcas registradas de Microsoft Corporation. **MSX WRITE** es marca registrada de ASCII Corporation.
• **CP/M** es marca registrada de Digital Research. **MSX-LOGO** es marca registrada de Logo Computer Systems Inc.

Los datos y especificaciones que figuran en este aviso pueden ser modificados sin previo aviso.



Director
Fernando Flores
Secretario de Redacción
Ariel Testori
Redacción
Fernando Pedró
Arte y Diagramación
Boccardo Diseño Gráfico
Servicios Fotográficos
Oscar Burriel y Claudio Veloso

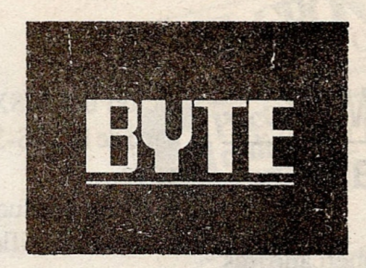
EDITORIAL PROEDI

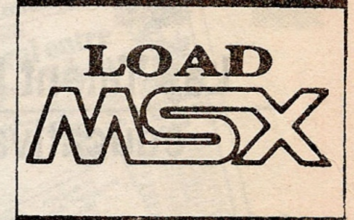
Presidente
Ernesto del Castillo
Vicepresidente
Cristian Pusso
Director Titular
Armengol Torres Sabaté
Director Suplente
Javier Campos Malbrán

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial Proedi S.A., México 625, 3º piso, (1097) Buenos Aires. Tel.: 30-0991 y 30-0200. Registro Nacional de la Propiedad Intelectual: E.T. Marca Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual. Todos los derechos reservados. ISSN 0326-8241. Impreso en Impresiones Gráficas Tabaré S.A.I.C., Erézcano 3158, Capital. Fotocromo de tapa: Columbia. Los ejemplares atrasados se venderán al precio del último número en circulación. Prohibida la reproducción total o parcial de los materiales publicados, por cualquier medio de reproducción gráfico, auditivo o mecánico, sin autorización expresa de los editores. Las menciones se realizan con fines informativos y técnicos, sin cargo alguno para las empresas que comercializan y/o los representan. Al ser informativa su misión, la revista no se responsabiliza por cualquier problema que pueda plantear la fabricación, el funcionamiento y/o aplicación de los sistemas y los dipositivos descriptos. La responsabilidad de los artículos firmados corresponde exclusivamente a sus autores. Distribuidor en Capital: Martino, Juan de Garay 358, P.B., Capital. Distribuidor en el interior: D.G.P., Hipólito Yrigoyen 1450, Capital, Tel.: 38-9266/9800.









EN LA ERA DE LAS FUSIONES K 64 Y LOAD MSX CONCENTRAN ESFUERZOS

Las cambiantes condiciones de los mercados obligó a grandes "casamientos" entre marcas y empresas importantes a nivel internacional. En el sector informático vimos sorprendentes uniones, que buscaban mayor eficiencia y mejor atención del cliente, para competir con productos de más calidad.

Nuestra editorial no permaneció ajena a esas transformaciones. Así fue que la revista "K 64. COMPUTACION PARA TODOS" incorporó hace un año el suplemento BYTE, con derechos exclusivos de la publicación de mayor difusión en los Estados Unidos en el campo de la computación personal.

Ahora, en su quinto año de vida, K 64 se propone incorporar el material de LOAD TALENT MSX, como una respuesta al desafío que implica ser editor hoy en la Argentina. Cambiamos para aumentar la productividad y así ofrecer un mejor servicio a nuestros lectores.

En el Suplemento Load Talent los usuarios de la norma MSX encontrarán las notas, informaciones y programas que hallaban en la revista que hoy termina su ciclo. Pero por el mismo precio también contarán con los informes exclusivos de BYTE, con el panorama de la informática en los centros más importante del mundo, y todos los datos de la realidad nacional.

Agradecemos a quienes nos acompañaron en este camino, y los invitamos a que continuen con nosotros a través de las páginas de K64, a partir del número aniversario que aparecerá el 3 de julio próximo. Hasta entonces.

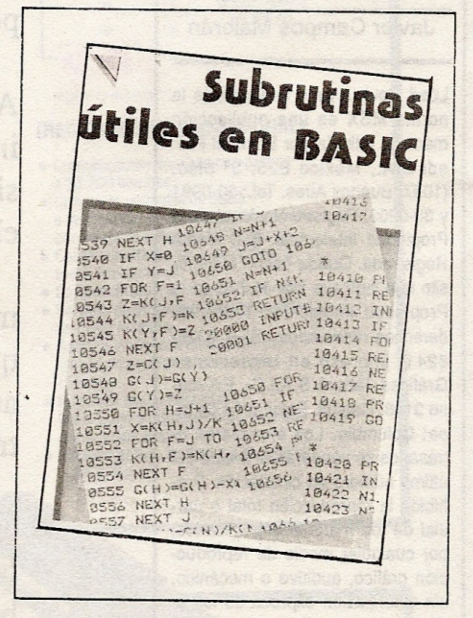
El Editor

Las Talent MSX educativas

Telemática S.A. lanzó al mercado tres nuevas computadoras orientadas para el uso en el área educativa.Los modelos se comercializan a pedido.
La Talent MSX DPS-201 es la computadora escolar secundaria MSX.
Viene con el teclado numérico TNX-410 y el sistema lo reconoce sin tener que cargar ningún software externo.
Cabe recordar que el teclado numérico es una unidad separada que dispone las teclas en una configuración habitual en las calculadoras.
En resumen, este kit se diferencia de la Talent DPC-200 MSX en que admite directamente el teclado numérico.
La Talent MSX2 TPP-311 es considerada la computadora escolar primaria MSX2.
Esta configuración es ideal para los talleres de Logo, ya que incorpora el Talent MSX LOGO en la computadora. Es decir, la computadora viene con el MSX BASIC versión 2.1 y en lugar de los accesorios y el Turbo BASIC, se le incorporó el LOGO.
Para invocar el LOGO basta con pulsar la barra espaciadora al encender la computadora.
Por otra parte, esta configuración incluye sólo 64 kbytes de RAM principal y 64 kbytes de Video RAM, por lo tanto no se pueden utilizar los modos de pantalla SCREEN 6, 7 y 8.
Finalmente la Talent MSX2 TPS-312 es tenida como la computadora escolar secundaria MSX2.
Esta computadora comparte con la TPS-311 la configuración de memoria (64 kbytes de RAM y 64 kbytes de Video RAM). Se reemplazaron el Turbo BASIC y los accesorios por el MSX Write y el MSX Plan.
Como en el ámbito secundario se está haciendo cada vez más necesario poder utilizar una planilla de cálculos o un procesador de textos, la incorporación del MSX Write y MSX Plan hace de la TPS-312 una opción interesante.
Como se producen a pedido, no tienen la sigla identificadora que las representa.

Subrutinas útiles en BASIC

La mayoría de los programas de aplicación, tal es el caso de los de gestión, utilizan rutinas similares. Por ejemplo, las pantallas con menú de opciones son rutinas que se diferencian solamente por las elecciones que permiten unas y otras.
Estas rutinas se pueden estandarizar y de este modo escribir un programa más corto.
Los objetivos del libro "Subrutinas útiles en BASIC" de Ian Sinclair (Edita Paraninfo) son enseñarnos a programar en bloques y al mismo tiempo facilitarnos rutinas aplicables a un sin fin de programas.
Debido a que el BASIC no es un lenguaje fosilizado, sino que por el contrario continuamente nos enteramos de versiones más desarrolladas, las rutinas se escribieron en un modo simple de BASIC, y emplean principalmente aquellas instrucciones que se encuentran en la mayoría de las pequeñas computadoras.
Sin embargo, lo dicho con anterioridad no garantiza que estas subrutinas funcionen en cualquier máquina. El programador deberá hacer los cambios cuando sea necesario.
Las rutinas se han escrito en BASIC de Microsoft (ideal para los usuarios de MSX), y constan de un comentario completo sobre cómo opera, de modo que el usuario podrá utilizarlas o cambiar las sentencias según la versión BASIC de su computadora.
El libro se divide en capítulos y cada uno de ellos abarca un tipo de subrutinas.
Para que no se tenga dificultad en utilizar estas subrutinas, todas se llaman con la instrucción GOSUB (número de línea) y se terminan con la sentencia RETURN.
Sólo faltaría describir brevemente algunas de las rutinas que encontraremos en estas páginas. Las de presentación visual, por ejemplo, convierten un título en intermitente o lo desplazan por la pantalla; las de técnica de entrada permiten tabular, formatear o comprobar el ingreso de los datos; las de búsqueda y especificación sirven para encontrar datos en listas o vectores y para ordenar la información según los diferentes métodos (Shell, rápida, etcétera).
Las subrutinas que pueden hacer operaciones útiles con vectores y matrices tales como multiplicación de elementos e impresión de una matriz



NxM no se han dejado a un lado.
El siguiente capítulo encierra algunas rutinas sobre gráficos y diagramas como subrayado, caracteres desplazables, etcétera.
Por último, hallaremos rutinas para guardar los datos en casete.
El libro es una referencia útil para los usuarios del BASIC, especialmente para los nuevos, y proporcionará algunas ideas que cada uno puede desarrollar cuando las necesite.

El Cronista de los domingos. Otro éxito diario.

Sí, El Cronista Comercial de los domingos
es el diario importante que usted merecía.
Resérvelo desde hoy.
Y asóciese al éxito.



El Cronista

Toda la información del sábado y domingo, cartelera de espectáculos, deportes, el gran editorial de la semana, y, una vez por mes, el periódico de perspectivas internacionales "World Paper".

Segunda Sección

El resumen de todo lo sucedido.
La semana día a día.
Los sucesos nacionales e internacionales que convulsionaron las primeras planas.
Y nuevas propuestas científicas, técnicas y económicas.

Tercera Sección

Economía. La evolución de los mercados.
El análisis. La reflexión.
La información pormenorizada que necesita conocer para decidir mejor.

Life & Fitness

La revista para toda la familia, con temas que interesan a la mujer de hoy.
Páginas a todo color. Fitness. Moda. Turismo. Cine. Artes Sociales. Decoración...
Para vivir y disfrutar de un modo más excitante su fin de semana.

La revista de "Alf"

16 páginas con las aventuras en historieta del más sorprendente personaje de la TV mundial.
Para deleite de los más pequeños.

Doble Chance

Su nueva oportunidad de ganar fabulosos premios gratis.
Encuentre las bases y la tarjeta para participar cada semana en su ejemplar de El Cronista del próximo domingo.

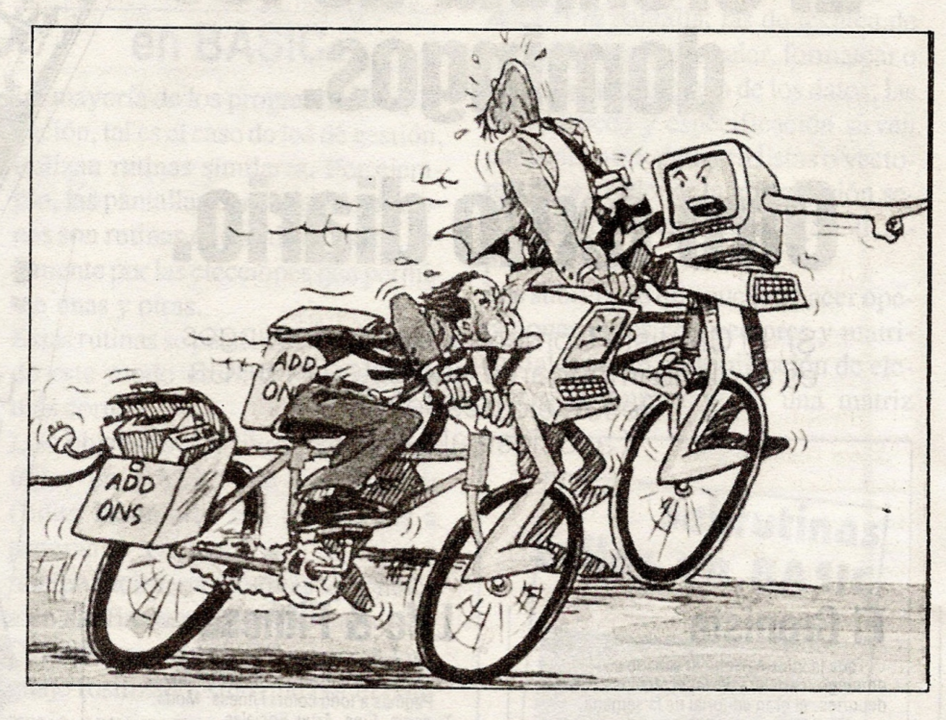
EL CRONISTA Comercial

Un éxito diario

UN BASIC VELOZ

El Turbo BASIC viene incorporado en la Talent MSX2 TPC-310, que toma el nombre de Turbo por el compilador. Sin embargo, funciona tanto en MSX1 como en MSX2..

El Turbo Basic es un compilador del lenguaje MSX BASIC para uso en computadoras personales MSX. La gran flexibilidad de diseño de la MSX permite agregar comandos al intérprete BASIC del sistema, con los que se tiene acceso directo al compilador Turbo, sin necesidad de edición y ensamble previo.



SOBRE COMPILADORES E INTERPRETES

Toda computadora digital maneja, por definición, exclusivamente códigos binarios, que son interpretados por la CPU (Central Processing Unit, Unidad Central de Procesamiento) para ejecutar comandos muy sencillos de manipulación de datos y cálculos elementales tales como suma y resta. Estos códigos se representan internamente mediante largas cadenas de unos y ceros.

Sin embargo, la computadora MSX, por ejemplo, maneja muy bien instrucciones tales como:

PRINT 3*SQR(4)/36

¿De dónde surge esta capacidad de por sí aparentemente incompatible con el funcionamiento de la CPU? De un programa incorporado en la computadora que se denomina Intérprete MSX BASIC.

Es un programa en lenguaje de máquina cuya tarea consiste en traducir al lenguaje máquina las instrucciones que recibe en BASIC. Una vez que las instrucciones están en código de máquina, la CPU ya puede trabajar con ellas y, por lo tanto, entenderlas.

De lo dicho surge que la CPU no entiende realmente BASIC, por lo que antes de que pueda ejecutar programas escritos en dicho lenguaje hay que traducirle todo a lenguaje de máquina. Es como cuando alguien nos habla, por ejemplo, en francés y no entendemos el idioma, para poder comunicarnos con él necesitamos o bien un buen diccionario, o bien un intérprete que nos vaya traduciendo el francés al castellano.

Análogamente, la parte del sistema operativo que realiza ese trabajo de traducción se denomina intérprete del BASIC. La "traducción" se realiza cada vez que se ejecutan las sentencias. El tiempo necesario para realizar esta traducción se suma al tiempo de ejecución del proceso especificado por las sentencias.

Ya definido qué es un intérprete, pasamos a analizar qué es un compilador, y sus ventajas frente al intérprete.

COMPILADOR VS. INTERPRETE

El hecho de que se tengan que traducir las instrucciones en BASIC antes de poder trabajar con ellas explica el porqué es más lento utilizar programas escritos en BASIC que los escritos en lenguaje de máquina, por ejemplo. El proceso de traducir un lenguaje a otro lleva tiempo, y a menudo las instrucciones en código máquina resultantes no son tan eficientes para el tipo de función para la que se diseñó.

Sin embargo, un programa escrito en lenguaje de máquina posee varias desventajas frente al intérprete. Las más destacadas son las siguientes:

a) Los programas en lenguaje de má-

quina son difíciles de leer y en ellos se complica la detección de los errores que tengan.

b) Son específicos para cada tipo de computadora (no son "portables").

c) Se utiliza una gran cantidad de instrucciones.

d) Los programas que usan aritmética complicada son difíciles de programar.

Por eso, como solución intermedia surge el compilador.

El proceso de traducción de un programa BASIC a código de máquina utilizando el intérprete se realiza instrucción por instrucción. Así, en el siguiente programa:

```
10 FOR I=1 TO 100
20 PRINT I;
30 NEXT I
```

la línea 20 es traducida 100 veces, pues aunque ya se tradujo PRINT I, la siguiente vez que se ejecuta, el intérprete "olvida" su traducción previa.

El compilador, en cambio, cumple una tarea similar a la del intérprete, pero la realiza una única vez.

Como el intérprete, el compilador traduce una tras otra las instrucciones BASIC. Pero la gran diferencia está en que el compilador traduce todo el programa en una sola operación, generando una versión completa del mismo en código de máquina que almacena como versión "objeto".

Siguiendo con el ejemplo anterior, el compilador traduce la instrucción PRINT I; una sola vez, y evita tener que "volver a traducirla" 99 veces.

Sin embargo, todo tiene su precio. La ventaja de hacer una traducción única en un compilador tradicional se diluye al tener que realizar varios pasos hasta lograr que un programa pueda ejecutarse.

Los pasos que se deben seguir en un compilador tradicional son éstos:

a) Utilizar un editor de texto para cargar el programa fuente y grabarlo (generalmente en disco).

b) Correr el programa compilador que analizará el programa fuente e indicará si hubo errores. En este caso, deberemos ir al paso a) nuevamente, y tantas veces como sean necesarias, hasta que no haya errores de compilación (por ejemplo, errores de sintaxis).

c) Una vez que el compilador tradujo exitosamente el programa fuente, genera otro archivo denominado "código intermedio", que es un programa que está más cerca del código de máquina. Pero todavía faltan incorporar ciertas definiciones ("Linkedición" en lenguaje informático. Consiste en resolver en direcciones de memoria definitivas las que genera el compilador, generalmente no fijadas sino de manera relativa).

d) Una vez ensamblado el programa, se ejecuta el mismo para verificar que no haya errores de lógica. En caso de existir, se deberá corregir el programa fuente y repetir todo el proceso de compilación hasta lograr que se ejecute correctamente.

Es decir que para el usuario del programa es mejor el compilador por su mayor eficiencia y velocidad, pero el programador debe pasar la "odisea de la compilación" si utiliza un compilador tradicional.

TURBO BASIC vs. COMPILADOR TRADICIONAL

El Turbo BASIC es un compilador que se puede invocar desde un programa en MSX BASIC.

La principal ventaja que tiene el compilador Turbo BASIC versus los compiladores tradicionales es que tiene la flexibilidad de un intérprete para edición y depuración (corrección) de programas, con la velocidad de los programas compilados.

Esto se logra gracias a la gran flexibilidad del sistema MSX, pues permite agregar comandos al intérprete que acceden al compilador en forma directa, sin tener que pasar por el paso intermedio de la edición y ensamble.

Con el Turbo BASIC, basta ingresar el programa como se hace habitualmente en la MSX, y luego, ejecutando un comando, el Turbo BASIC toma el control, traduce todo el programa y lo ejecuta, con la única -y gran- diferencia de la velocidad de ejecución.

COMO UTILIZAR EL TURBO BASIC

Ahora vamos a describir algunas de las técnicas de programación que se utilizan con el compilador BASIC.

Existen dos formas de emplear el compilador BASIC:

a) Compilar todo el programa BASIC en una sola operación y ejecutarlo.

b) Durante la ejecución normal del programa BASIC, compilar y ejecutar las líneas comprendidas entre los comandos CALL TURBO ON y CALL TURBO OFF. La ejecución del programa BASIC continúa a partir de la instrucción siguiente al CALL TURBO OFF.

a) SI SE COMPILA LA TOTALIDAD DEL PROGRAMA

En este caso se programa en forma normal en BASIC (con algunas restricciones que luego detallaremos). Luego simplemente con la instrucción CALL RUN (o _RUN), se compila todo el programa y se lo ejecuta. El tiempo de ejecución es de 20 a 30 veces menor que el del MSX-BASIC.

b) SI SE COMPILA PARTE DEL PROGRAMA

Esta opción se utiliza en caso de ser imprescindible emplear alguna sentencia de intérprete BASIC que este compilador no acepte (tal como manejo de archivos).

En este caso, a las líneas de programa que se quieran compilar se les antepone una línea con la sentencia CALL TURBO ON (o _TURBO ON) y al final de las líneas a compilar se les coloca la sentencia CALL TURBO OFF (o _TURBO OFF)

Hay que tener en cuenta que estas sentencias deben ser únicas en la línea, o sea que no se pueden utilizar multisentencias en las líneas en que figuren CALL TURBO ON o CALL TURBO OFF.
Finalmente, se ejecuta la sentencia RUN y el programa comenzará su ejecución. Cuando el intérprete BASIC encuentra la sentencia CALL TURBO ON, le pasa el control al compilador TURBO BASIC, que se encarga de compilar todo el programa hasta que encuentra la sentencia CALL TURBO OFF y ejecuta las rutinas compiladas.
Cuando termina de ejecutar el bloque compilado, el TURBO BASIC devuelve el control al intérprete BASIC y éste continúa la ejecución de las sentencias siguientes al CALL TURBO OFF, en forma normal.
Hay algunos puntos que debemos destacar:
a) Las variables utilizadas dentro de las rutinas compiladas desaparecen al terminar su ejecución.
b) Las variables fuera del bloque compilado son totalmente independientes de las variables dentro del bloque, aun si son homónimas.
Sin embargo, muchas veces necesitamos que la parte compilada "conozca" las variables de la parte interpretada, para lo cual deberemos pasar parámetros entre ambas partes.
Esto se puede interpretar como que el contenido de ciertas variables se hace coincidente tanto dentro como fuera del bloque de compilación. Es decir que estas variables son compartidas por el Turbo BASIC y el intérprete.
En el Turbo BASIC, se puede lograr esto sólo para variables enteras simples u ordenamientos (arrays) del mismo tipo con subíndice.
Para pasar parámetros al compilador, basta escribir de CALL TURBO ON; la sentencia CALL TURBO ON (XX,XX,XX), donde XX son nombres de variables que van a ser compartidas entre la parte compilada y la interpretada.
Por ejemplo:
CALL TURBO ON(A%,B( ))
En este caso, la variable A% y el vector B( ) contienen los mismos valores, tanto dentro como fuera de la parte compilada del programa.

TIPOS DE VARIABLES Y PRECISION

El tipo de variable numérica por defecto es punto flotante. A diferencia del intérprete, la representación del número se hace mediante 3 bytes. Pueden tomar valores desde 2.939E-39 a 1.7013E+38, y la precisión de los cálculos es de 4.5 cifras significativas.
La representación interna es la siguiente:
<exponente><mantisa menos significativa> <mantisa más significativa>
Ejemplo:
0.5 -> 80h,00h,00h
-1 -> 81h,00h,80h
Turbo BASIC maneja la aritmética de enteros en forma análoga al MSX BASIC.
Los nombres de las variables pueden ser de cualquier longitud, pero sólo los dos primeros caracteres son significativos. Los ordenamientos pueden tener cualquier dimensión.
Debe tenerse en cuenta que no se pueden realizar operaciones con variables alfanuméricas muy complejas. Para estos casos es recomendable hacerlo fuera del bloque de compilación.

DIRECTIVAS ESPECIALES DEL TURBO BASIC

Mediante directivas especiales se puede indicar al Turbo BASIC que realice tareas especiales que no están previstas por el intérprete. Las directivas se componen de una línea que comienza con "comentario" (REM o ') seguida de la directiva. Estas directivas son las siguientes:

a) DIRECTIVA #i

Cuando se encuentra '#i' al principio de una línea de comentario, los valores ubicados a continuación en la misma línea se ubican directamente en el código objeto (byte por byte). Un número precedido del símbolo "@" especifica la dirección de una línea de programa (palabra de 2 bytes), y un nombre de variable especifica la dirección de la variable (palabra de 2 bytes).
Por ejemplo:

```
10 A%=0
20 IF A%>100 THEN END
30'#I&H2A, A%, &H23, &H22, A%, &HC3, @10
```

En este caso, la línea 30 es completamente idéntica a:

```
30 A%=A%+1:GOTO 10
```

Obsérvese que los valores indicados en la línea 30 en hexadecimal corresponden al código máquina del Z80 y, por lo tanto, para usar esta directiva es necesario conocer el tema.
En el ejemplo, en lenguaje ensamblador la línea 30 se habría expresado:

```
LD HL,(A%)
INC HL
LD (A%),HL
JP @10
```

b) DIRECTIVA #c

La directiva "#c" se utiliza para habilitar/inhabilitar el ajuste de coordenadas. Cuando se especifica "#c+", la coordenada Y (vertical) se ajusta si se produce desborde. "#c-" suprime el ajuste. El valor por defecto es "#c+".
Ejemplo:

```
10 SCREEN 2
20 REM #C-
30 LINE (0,0)-(255,255)
```

Continúa en pág. 10

SENTENCIAS BASIC Y TURBO BASIC

ABS Igual que MSX-BASIC
AND Igual que MSX-BASIC
ASC Igual que MSX-BASIC
ATTR$ N.D. (No Disponible)
ATN Igual que MSX-BASIC
AUTO N.D.
BASE N.D.
BEEP Igual que MSX-BASIC
BIN$ Igual que MSX-BASIC
BLOAD N.D.
BSAVE N.D.
CALL N.D.
CDBL N.D.
CHR$ Igual que MSX-BASIC
CINT N.D.
CIRCLE No se admite <excen-tricidad> ni <ángulo inicial/final>.
CLEAR N.D.
CLOAD N.D.
CLOSE N.D.
CLS Igual que MSX-BASIC
CMD N.D.
COLOR Igual que MSX-BASIC
CONT N.D.
COPY Sólo se admite COPY de gráficos (en MSX2).
COS Igual que MSX-BASIC
CSAVE N.D.
CSNG N.D.
CSRLIN Igual que MSX-BASIC
CVD N.D.
CVI N.D.
CVS N.D.
DATA Igual que MSX-BASIC
DEF Sólo se acepta 'DEF USR'.
DEFDBL Igual que MSX-BASIC
DEFINT Igual que MSX-BASIC
DEFSNG Igual que MSX-BASIC
DEFSTR Igual que MSX-BASIC
DELETE N.D.
DIM Igual que MSX-BASIC, con la diferencia que DIM debe ejecutarse antes de cualquier sentencia ejecutable, y la declaración debe realizarse utilizando constantes enteras.
DRAW N.D.
DSKF N.D.
DSKI$ N.D.
DSKO$ N.D.
ELSE Igual que MSX-BASIC
END Igual que MSX-BASIC
EOF N.D.
EQV N.D.
ERASE N.D.
ERL N.D.
ERR N.D.
ERROR N.D.
EXP Igual que MSX-BASIC
FIELD N.D.
FILES N.D.
FIX Igual que MSX-BASIC
FN N.D.
FOR Igual que MSX-BASIC
FPOS N.D.
FRE N.D.
GET N.D.
GO TO Igual que MSX-BASIC
GOSUB Igual que MSX-BASIC
GOTO Igual que MSX-BASIC
HEX$ Igual que MSX-BASIC
IF Igual que MSX-BASIC
IMP N.D.
INKEY$ Igual que MSX-BASIC
INP Igual que MSX-BASIC
INPUT Una sola variable por INPUT.
INSTR Igual que MSX-BASIC
INT Igual que MSX-BASIC
IPL N.D.
KEY Igual que MSX-BASIC, excepto 'KEY <n>, <alfanuméri-co>' y 'KEY LIST'
KILL N.D.
LEFT$ Igual que MSX-BASIC
LEN Igual que MSX-BASIC
LET Igual que MSX-BASIC
LFILES N.D.
LINE Igual que MSX-BASIC, excepto 'LINE INPUT'.
LIST N.D.
LLIST N.D.
LOAD N.D.
LOC N.D.
LOCATE Igual que MSX-BASIC, excepto que deben indicarse simultáneamente x e y mientras que el <activador de cursor> no se admite.
LOF N.D.
LOG Igual que MSX-BASIC
LPOS Igual que MSX-BASIC
LPRINT Igual que MSX-BASIC
LSET N.D.
MAX N.D.
MERGE N.D.
MID$ Igual que MSX-BASIC
MKD$ N.D.
MKI$ N.D.
MKS$ N.D.
MOD Igual que MSX-BASIC
MOTOR N.D.
NAME N.D.
NEW N.D.
NEXT Siempre necesita especificación de variable.
NOT Igual que MSX-BASIC
OCT$ Igual que MSX-BASIC
OFF Igual que MSX-BASIC
ON Igual que MSX-BASIC, excepto 'ON STOP GOSUB'.
OPEN N.D.
OR Igual que MSX-BASIC
OUT Igual que MSX-BASIC
PAD Igual que MSX-BASIC
PAINT Igual que MSX-BASIC
PDL Igual que MSX-BASIC
PEEK Igual que MSX-BASIC
PLAY N.D.
POINT Igual que MSX-BASIC
POKE Igual que MSX-BASIC
POS Igual que MSX-BASIC
PRESET Igual que MSX-BASIC
PRINT Igual que MSX-BASIC, excepto que SPC y TAB no se aceptan, el espaciado por zonas (';') actúa diferente, y los números no se alinean para conformar la línea (es decir, no se envían al siguiente renglón si exceden al mismo, simplemente se cortan y siguen abajo).
PSET Igual que MSX-BASIC
PUT Sólo se admite 'PUT SPRITE'.
READ Igual que MSX-BASIC
REM Igual que MSX-BASIC
RENUM N.D.
RESTORE Igual que MSX-BASIC
RESUME N.D.
RETURN Igual que MSX-BASIC
RIGHT$ Igual que MSX-BASIC
RND Igual que MSX-BASIC
RSET N.D.
RUN Igual que MSX-BASIC, excepto que las variables no se inicializan.
SAVE N.D.
SCREEN Sólo se admite <modo pantalla> y <tamaño sprites>.
SET Sólo se admite SET PAGE (en MSX 2).
SGN Igual que MSX-BASIC
SIN Igual que MSX-BASIC
SOUND Igual que MSX-BASIC
SPACE$ Igual que MSX-BASIC
SPC(N.D.
SPRITE Igual que MSX-BASIC
SQR Igual que MSX-BASIC
STEP Igual que MSX-BASIC
STICK Igual que MSX-BASIC
STOP Igual que MSX-BASIC
STR$ Igual que MSX-BASIC
STRIG Igual que MSX-BASIC
STRING$ Igual que MSX-BASIC
SWAP Igual que MSX-BASIC
TAB(N.D.
TAN Igual que MSX-BASIC
THEN Igual que MSX-BASIC
TIME Igual que MSX-BASIC
TO Igual que MSX-BASIC
TROFF N.D.
TRON N.D.
USING N.D.
USR Igual que MSX-BASIC
VAL Igual que MSX-BASIC
VARPTR Igual que MSX-BASIC
VDP Igual que MSX-BASIC
VPEEK Igual que MSX-BASIC
VPOKE Igual que MSX-BASIC
WAIT Igual que MSX-BASIC
WIDTH N.D.
XOR Igual que MSX-BASIC
' Igual que MSX-BASIC
> Igual que MSX-BASIC
= Igual que MSX-BASIC
< Igual que MSX-BASIC
+ Igual que MSX-BASIC
- Igual que MSX-BASIC
* Igual que MSX-BASIC
/ Igual que MSX-BASIC
^ Igual que MSX-BASIC
\ Igual que MSX-BASIC

```
40 '#C+
50 LINE (0,0)-(255,255)
```

El ajuste de coordenadas consiste en verificar los límites de la pantalla respecto de las coordenadas indicadas por el usuario. Si se exceden dichos límites, el ajuste consiste en ignorar las coordenadas fuera de rango.

c).DIRECTIVA #n

La directiva "#n" activa la verificación de desborde de números enteros, cuando, por ejemplo, se hace un ciclo FOR-NEXT con variables enteras, de la forma:

```
10 FOR I%=0 TO &H7FFF
20 NEXT I%
```

Bajo el entorno del MSX BASIC, esta rutina generará un error de "desborde", pero si se la ejecuta con el Turbo BASIC, entrará en un loop infinito.

Para evitar este inconveniente, se debe incorporar la directiva especial "#n+", que le indica al compilador que vigile si los cálculos se van fuera del rango de los enteros.

Sin embargo, es conveniente no abusar de esta opción, ya que consume más memoria y disminuye la velocidad de ejecución.

Para cancelar esta directiva, se debe colocar "#n-".

MANEJO DE INTERRUPCIONES: SENTENCIAS ON... GOSUB ...

El Turbo BASIC soporta el manejo de interrupciones. Sin embargo, cuando se utilizan estas rutinas se genera un código compilado largo y lento, debido a que la rutina de verificación de interrupción es invocada cada vez que se ejecuta una sentencia.

El manejo de alfanuméricos se mantiene igual al MSX BASIC, con ciertas limitaciones que se describen a continuación.

DIFERENCIAS ENTRE TURBO BASIC Y MSX BASIC

Veamos algunas observaciones que se deben tener en cuenta al programar en Turbo BASIC.

1) Todos los ordenamientos deben ser dimensionados previamente, aun los que vayan a utilizar subíndices menores a 10.

2) Las sentencias DIM sólo pueden estar precedidas por las sentencias DEFINT, DEFSNG, DEFDBL, REM, DATA y DIM; de lo contrario se genera el error "DIM ya usado" (Redimensioned array).

3) Cuando se llame a rutinas en código de máquina, los parámetros que van a continuación de la sentencia USR sólo pueden ser valores enteros.

4) Cada variable alfanumérica ocupa 256 bytes de memoria, por lo tanto su uso indiscriminado puede completar la memoria rápidamente.

5) Las variables que se utilizan como distribuidor en las sentencias ON GOTO y ON GOSUB son tomadas para la ejecución con el valor que surge del resto de la división de las mismas por 256.

6) No se verifican los límites de los parámetros que se pasan a las funciones ni los subíndices de los ordenamientos.

7) La precisión numérica es de hasta 4,5 dígitos, y el rango numérico va desde +2,939E-39 a +1,7012E+38.

8) Cuando se impriman números con la sentencia PRINT, las cifras mayores a 10000 serán puestas en notación científica (con el exponente E).

9) Dentro del Turbo BASIC, tanto las variables de doble como de simple precisión son tratadas de igual forma, por lo tanto su distinción dentro del programa (por ejemplo A% con A!) no es válida.

10) Los resultados de operaciones que no sean divisiones y potencias normalmente son de tipo entero. Esto puede traer la siguiente consecuencia:

```
10 A%=100
20 PRINT A%*A%
```

En MSX BASIC, el resultado será el número real 40000, pero en el Turbo se obtendrá el número entero -25536. Se puede evitar multiplicando todo por 1! (variable real).

11) No se hace ningún tipo de validación de subíndices para los ordenamientos, excepto en el número de dimensiones.

12) No se verifican los rangos de los parámetros en _TURBO ON.

13) Se debe especificar la variable en la sentencia NEXT (en esta versión del Turbo BASIC).

Ejemplo:

10 FOR I=1 TO 10:NEXT I (correcto)

10 FOR I=1 TO 10:NEXT (incorrecto para Turbo BASIC).

14) Sólo se admite una variable por sentencia INPUT.

15) Los números de línea en la directiva #i no se renumeran.

En la tabla 1 se resumen las sentencias MSX BASIC soportadas por el Turbo BASIC.

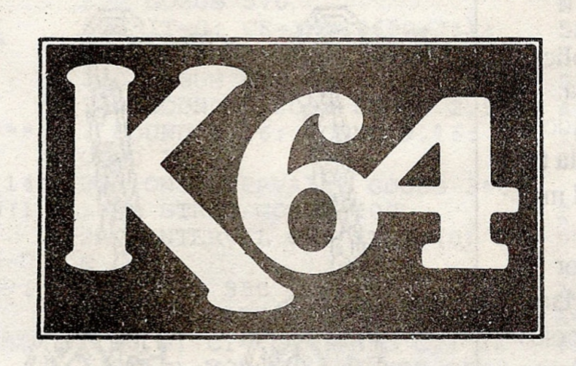
CONCLUSIONES

Este compilador es una interesante solución de compromiso entre un compilador "completo" (que no tenga excepciones respecto al intérprete) y su facilidad de utilización desde el MSX BASIC. Hemos sintetizado el contenido del manual de uso del Turbo BASIC por considerar que es lo suficientemente explícito como para mostrar a los no usuarios de la Talent MSX2 Turbo las bondades del Turbo BASIC. Como conclusión podemos esperar, para los usuarios de la MSX1, su aparición en el mercado como cartucho, ya que en vista a su performance significa un verdadero avance para la norma MSX.

Hugo Daniel Caro

30

Becas de trabajo



Convoca a jóvenes estudiantes o recién egresados de carreras de computación o electrónica, de Capital Federal y del interior del país, quienes serán seleccionados por la empresa Telemática S.A., productora de las computadoras Talent MSX, para realizar trabajos en el área de promoción.

Los interesados deben escribir a :
"K64. Computación para todos"
Becas de Trabajo
México 625, 3º piso
(1097) Buenos Aires

METRONOMO MSX

Tipo: Utilitario
Autora: Claudia Jiménez

Seguimos descubriendo aplicaciones para nuestra computadora. Esta vez es un utilitario destinado a los estudiantes de música. Simula un aparato usado por todos ellos: el metrónomo.

Este instrumento, inventado por el mecánico austríaco Leonardo Maelzel, se emplea para medir los diversos grados de velocidad del movimiento musical.

La simulación que alcanza este programa es realmente excelente. El usuario puede elegir entre varios tempos predefinidos: grave, largo, lento, adagio, larghetto, andante, andantino, moderato, allegretto, allegro, allegro assai, allegro vivacce, vivacce, presto y prestissimo. También es posible definir tempos intermedios y modificar el ritmo.

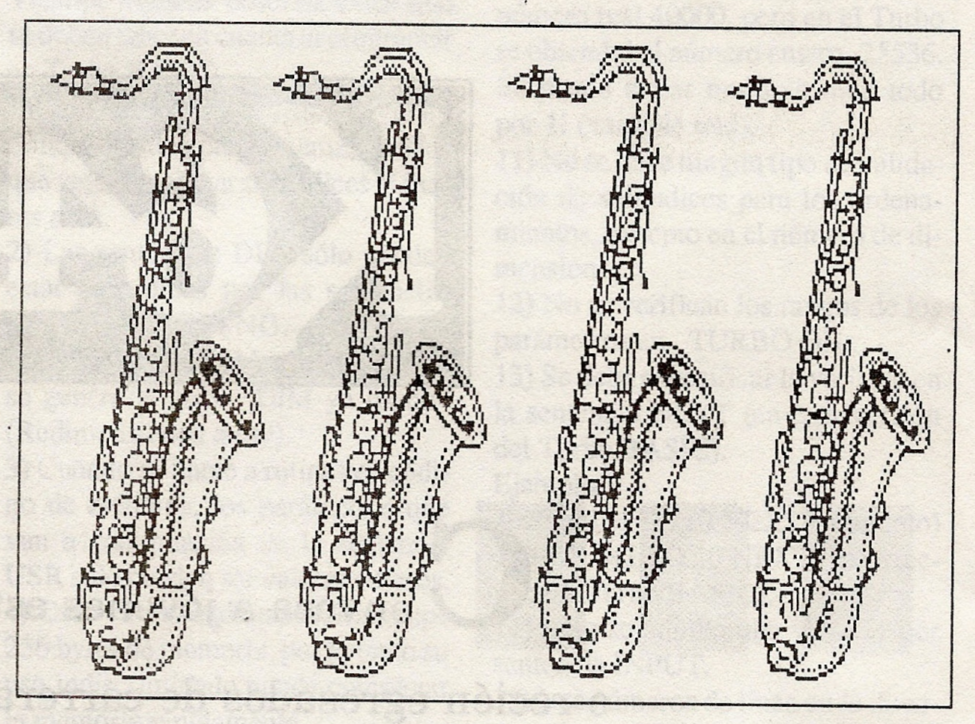
El programa comienza estableciendo un "andante", pero se puede modificar si se aprieta la barra espaciadora. Esto habilitará las teclas de los números 1 al 6. Con las mismas se efectúan los cambios. Para continuar bastará con presionar el número 7.

Pero este programa no solo sirve como una excelente ayuda para los estudiantes de música, sino que por sus características ejemplifica varios temas de programación.

Para comenzar, mencionemos la prolija presentación, en cajas, que le da un colorido muy especial.

Otro aspecto destacable es el uso de las interrupciones, manejadas con las instrucciones INTERVAL y STRIG.

Por supuesto que dados los fines del programa el sonido debe ser agradable. El contol del comando SOUND cumple muy bien esta función.



Finalmente llamamos la atención de los programadores sobre el movimiento pendular del metrónomo, logrado con una sencilla subrutina.

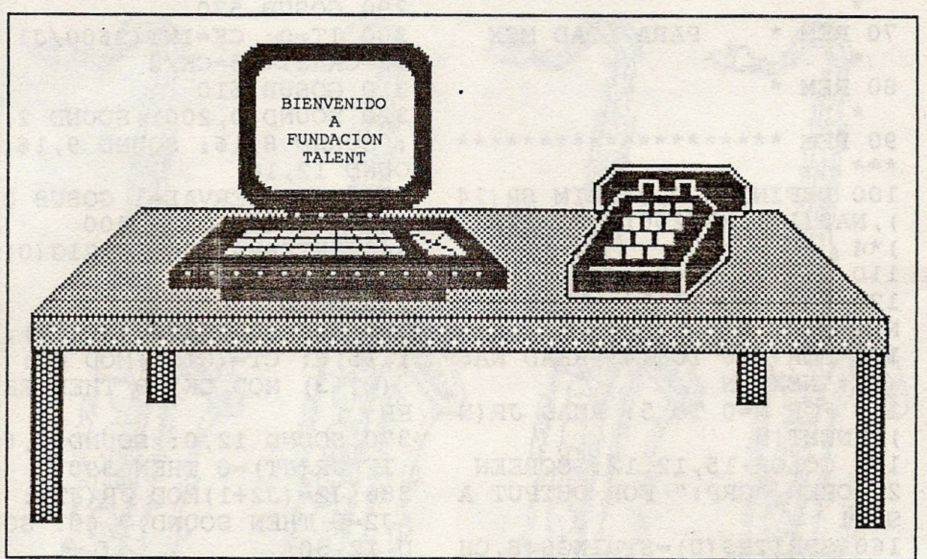
Por todas las características mencionadas este programa será de gran utilidad para todos los programadores y estudiantes de música.

Estructura del programa

10-90 Datos de la autora.
100 Define variables enteras. Dimensina vectores. Define PI.
110 Define valores por defecto.
120-140 Lee valores y nombres de tempos y ritmos.
150 Define pantalla para poder imprimir caracteres alfanuméricos en modo mediana resolución.
160 Define el diseño del sprite.
170-200 Crea cajas.
210-280 Imprime leyenda.
290 A la subrutina que imprime valores en cajas.
300 Define variables de tiempo.
310 A la subrutina que define intervalos del sprite.
320 Sonido de marca.
330 Pregunta estado de interrupción y de la barra espaciadora.
340 Activa interupciones y a la barra espaciadora como disparador de interrupción.
350 Línea de espera.
360 Mueve aguja del metrónomo.
370-390 Sonido intermedio.
400 Desactiva interrupciones y barra espaciadora.
410 A la subrutina que imprime valores en cajas.
420-560 Modifica valores según la opción ingresada.
570-600 Imprime valores en cajas.
610-650 Define intervalos del sprite.
660-700 Datos.

BBS Fundación Talent

Finalmente, se concretó un proyecto esperado por la comunidad educativa y los usuarios de la norma MSX: la creación de una BBS para los usuarios Talent.



Esta BBS funciona en el teléfono 38-9681. Es gratuita y de libre acceso en horarios de 10 a 18 hs., aunque, por el momento, no está conectada permanentemente (insistir hasta conseguir la comunicación).

Las principales características de la BBS son las siguientes:
Norma: CCITT
Baudios: 300 baudios
Longitud de palabra: 8 bits
Paridad: N
Stop bits: 1

En esta BBS se pueden dejar mensajes o recibirlos de otros usuarios, además de dejar y recuperar programas de dominio público.

A los usuarios de Talent, esta BBS les permitirá (modem mediante) obtener distintos programas de interés general, incluso los que hemos publicado en el Rincón del Usuario Talent.

A continuación mostramos una comunicación de un usuario que utiliza la BBS por primera vez.

BIENVENIDO A FUNDACION TALENT

```
Este BBS no tiene fines de lucro y
está a cargo de Telemática S.A.
Rogamos no utilizarlo en forma indis-
criminada.
Todas las entradas quedan registradas
y son verificadas posteriormente.

¿Cuál es su nombre? Jorge
¿Cuál es su apellido? Agnese
...chequeando usuarios...
¿Con qué tipo de sistema está llamando?
(Pulse <RETURN> si es MSX)
¿Cuál es su localidad y provincia ? La-
nús - Buenos Aires
Bienvenido a Fundación.
Este sistema está orientado a los
usuarios de computadoras MSX, espe-
cialmente dentro del área educativa.
Las reglas mínimas que todos tratamos
de respetar son:

1. Usar nuestro nombre real
2. Indicar nuestra localidad
3. Tratar de no llamar más de una vez
   por día para no dejar a otro afuera.
4. Usar el Correo para comunicarnos
   sin ofender a nadie.
   Gracias.
JORGE AGNESE de LANUS - BUENOS AIRES

<C>ambiar nombre o dirección
<D>esconectar (sin registro)
<R>egistrarse? r
Entre la CLAVE que usará para volver a
acceder? nene
Reingrese su CLAVE para su verificación
? ....
Por favor, RECUERDE su clave
¿SU TERMINAL PERMITE MINUSCULAS (S/N)?
s
MAYUSCULAS y minuscúsculas
GRAFICOS:    ¿<N>ada,<A>scii,<C>o-
lor,<H>elp? n
GRAFICOS: Nada
¿Transferencia   por   omisión   Tipo
```

```
10 REM ***********************
20 REM *                     *
30 REM *    METRONOMO MSX    *
40 REM *                     *
50 REM * POR CLAUDIA JIMENEZ *
60 REM *                     *
70 REM *    PARA LOAD MSX    *
80 REM *                     *
90 REM ***********************
100 DEFINT X,C,J: DIM SP(14),NA$(14),XP(200): PI=ATN(1)*4
110 J1=72: JN=5: JT=3: MN=0
120 FOR N=0 TO 14: READ SP(N): NEXT N
130 FOR N=0 TO 14: READ NA$(N): NEXT N
140 FOR N=0 TO 5: READ JR(N): NEXT N
150 COLOR 15,12,12: SCREEN 2: OPEN "GRP:" FOR OUTPUT AS #1
160 SPRITE$(0)=STRING$(8,CHR$(255))
170 LINE(40,30)-(216,191),8,BF
180 LINE(50,40)-(206,181),14,BF
190 LINE(76,70)-(186,80),1,BF
200 LINE(75,130)-(131,140),4,BF
210 PSET(81,50),14: PRINT #1,"METRONOMO MSX"
220 PSET(80,131),4: PRINT #1,"|"
230 PSET(80,133),4: PRINT #1,".="
240 PSET(72,113),12: PRINT #1,"3<<  TEMPO  >>4"
250 PSET(76,143),14: PRINT #1,"1<    >2"
260 PSET(146,143),14: PRINT #1,"5< >6"
270 PSET(146,155),14: PRINT #1,"RITMO"
280 PSET(80,170),14: PRINT #1,"7...CONTINUAR"
290 GOSUB 570
300 IT=0: CK=INT(3600/J1): C2=CK*2: C3=CK/2
310 GOSUB 610
320 SOUND 0,200: SOUND 2,50: SOUND 8,16: SOUND 9,16: SOUND 12,10
330 ON INTERVAL=1 GOSUB 360: ON STRIG GOSUB 400
340 INTERVAL ON: STRIG(0) ON
350 GOTO 350
360 PUT SPRITE 0,(XP(CT),71),15,0: CT=(CT+1)MOD C2: IF (CT+3) MOD CK >0 THEN RETURN
370 SOUND 12,0: SOUND 7,62: IF JR(JT)=0 THEN 390
380 J2=(J2+1)MOD JR(JT): IF J2=0 THEN SOUND 7,60: SOUND 12,50
390 SOUND 13,0: RETURN
400 INTERVAL OFF: STRIG(0) OFF
410 GOSUB 570
420 MN=0: KY$=INPUT$(1): X1=ASC(KY$)-48: IF X1<1 OR X1>7 THEN 420
430 ON X1 GOTO 440,460,480,490,500,510,520
440 J1=J1-1: IF J1<40 THEN J1=208
450 MN=1: GOTO 550
460 J1=J1+1: IF J1>208 THEN J1=40
470 MN=1: GOTO 550
480 JN=(JN+14)MOD 15: J1=SP(JN): GOTO550
490 JN=(JN+1)MOD 15: J1=SP(JN): GOTO550
500 JT=(JT+5)MOD 6: GOTO550
510 JT=(JT+1)MOD 6: GOTO550
520 IT=0: CK=INT(3600/J1): C2=CK*2: C3=CK/2
530 GOSUB 610
540 INTERVAL ON: STRIG(0) ON: RETURN
550 GOSUB 570
560 GOTO 420
570 LINE(60,100)-(196,110),4,BF: IF MN=0 THEN PSET(64,101),4: PRINT #1,NA$(JN)
580 LINE(100,130)-(131,140),4,BF: PSET(96,132),4: PRINT #1,J1
590 LINE(160,130)-(170,140),4,BF: PSET(155,131),14: PRINT #1,JR(JT)
600 RETURN
610 PSET(22,2),12: PRINT #1,"PULSE LA BARRA PARA CAMBIAR"
620 M=0:FOR N=0 TO PI STEP PI/CK
630 XP(M)=SIN(N)*50+128:XP(M+CK)=256-XP(M): M=M+1
640 NEXT N
650 RETURN
660 DATA 40,46,52,58,63,72,80,92,108,132,144,152,160,184,208
670 DATA GRAVE,LARGO,LENTO,ADAGIO,LARGHETTO,ANDANTE,ANDANTINO
680 DATA MODERATO,ALLEGRETTO,ALLEGRO,ALLEGRO ASSAI,ALLEGRO VIVACCE
690 DATA VIVACCE,PRESTO,PRESTISSIMO
700 DATA 0,1,2,3,4,6
```



<A>scii,em? a
PROTOCOL: Ascii
Nulos ¿(S/N)? s
Nulos Sí

Ingresa JORGE AGNESE

RBBS-PC VERSION CPC14.1D NODO1
OPERANDO A 300 BAUD,N,8,1

<Ctrl K> aborta. <Ctrl S> suspende

FUNCIONES SOPORTADAS

- Cuestionarios de salida y entrada.
- Soporte de gráficos, acentos y eñes.
- Envío y recepción de mensajes.
- Subida y bajada de archivos binarios.
- Boletines del Sysop.
- Velocidades de 300 a 9600 baudios.

Nro. de archivos bajados: 0
Nro. de archivos subidos: 0

Su PERFIL es el siguiente

(Puede cambiarlo en Utiles)

GRAFICOS: Nada
PROTOCOL: Ascii
MAYUSCULAS y minusc.
Line Feeds On
Nulos Sí
Campanilla On

¿Quiere saltear los 9 boletines? n

BOLETINES

1 La Fundación Talent
2 Objetivos de la BBS FUNDACION TALENT
3 Novedades sobre MSX
4 Lista de BBSs
5 Calendario de Actividades
6 Seminarios y Cursos
7 Grupos de interés en la BBS
8 Programa XMODEM: cómo obtenerlo
9 Programa XMODEM: cómo usarlo

Boletín Nro. 1 ... 9

¿L)ista, <RETURN> continúa? 2

<Ctrl K> aborta. <Ctrl S> suspende

BBS FUNDACION TALENT
====================

La BBS Fundación Talent es un Sistema de Correo y Boletines público administrado por la Fundación Talent (ver boletín Nro.1), para facilitar el intercambio de experiencias y opiniones entre establecimientos educativos.
El objetivo es que las distintas experiencias realizadas por las diferentes instituciones en el ámbito educativo no queden aisladas y puedan ser compartidas por todos aquellos que deseen participar.
Cualquier sugerencia para su uso, o cualquier aporte que usted pueda hacer será bienvenido.

Quedan 26 minutos.

Boletín Nro. 1 ... 9

¿L)ista, <RETURN> continúa?

Chequeando mensajes.

Lo siento, JORGE, no tiene correo.

RBBS-PC Versión CPC14.1D Nodo1

Llamada Nro. 87
Nro. de mensajes disponibles: 8
Siguiente Nro. de mensaje: 10

Quedan 25 minutos.

MENU PRINCIPAL

COMUNICACIONES

B)oletines	A)yuda
C)orreo personal	H)asta pronto
N)otas al Sysop	P)ágina
O)perador	U)tiles
S)istema archivos	F)unciones

MENSAJES

D)ejar mensajes	E)liminar msjs

L)eer mensajes M)irar mensajes
R)evisar mensajes

¿Menú Principal>? U

Quedan 23 minutos

=============== UTILES ===============

B)audios P)rotocolo
A)yuda M)argen de msjs
N)ueva clave S)tatus
U)suarios C)ampanilla
L)etras G)ráficos
O)pciones T)iempo
E)xperto V)uelve al Menú

¿Utiles>? v

Quedan 22 minutos

========== MENU PRINCIPAL ==========

COMUNICACIONES

B)oletines A)yuda
C)orreo personal H)asta pronto
N)otas al Sysop P)ágina
O)perador U)tiles
S)istema archivos F)unciones

MENSAJES

D)ejar mensajes E)liminar msjs
L)eer mensajes M)irar mensajes
R)evisar mensajes

¿Menú Principal>? S

Quedan 21 minutos

SISTEMA DE ARCHIVOS

B)ajar archivo S)ubir archivo
L)istar archivo N)ovedades
R)evisar archivo

AUXILIARES

V)olver al menú H)asta pronto
I)nformación A)yuda

¿Archivos>? l

<Ctrl K> aborta. <Ctrl S> suspende

DIRECTORIOS DISPONIBLES

Dir ñContenido ñFormato

LOGO ñProgramas Logo ñAscii
BASIC ñProgramas Basic ñAscii
SOFT ñSoft Educativo ñBin
UTILES ñUtilitarios ñBin
LENGUAJE ñIntérpretes y Comp ñBin

5 directorio(s) disponible(s)

Para consultar un directorio específico tipee, por ejemplo:

L;LOGO (RETURN)

Quedan 21 minutos.

SISTEMA DE ARCHIVOS

B)ajar archivo S)ubir archivo
L)istar archivo N)ovedades
R)evisar archivo

AUXILIARES

V)olver al menú H)asta pronto
I)nformación A)yuda

¿Archivos>? L;LOGO

AHORCADO 81771-01-84 Juego del ahorcado
ANIMAL 33931-01-84 Base de datos y clasificación de animales
CALAVERA 50741-01-84 Juego del salto de caballo
CONJUNTO 22451-01-84 Procedimientos de lógica de conjuntos
FRACCION 25221-01-84 Soft educativo de fracciones
FUNCIONE 27451-01-80 Graficador de funciones
HANOI 23091-01-80 Torres de Hanoi con actores y movimiento
LOGOALFA 21551-01-84 Ordenador alfabético de palabras
MEDIANAS 11891-01-84 Triángulos y sus medianas
NIM 8091-01-84 Juego del Nim

OREJA 32141-01-84 Música aleatoria
PRIMOS 83441-01-84 Generador de números primos
RAPIDO 35861-01-84 Graficador fácil de Logo
ROCK 12531-01-84 Música recursiva
SUMAVEC 9611-01-84 Suma de vectores
UTILES 21121-01-84 Utiles de Logo (while, pertenece?, etc.)
18 Archivos

Quedan 20 minutos.
======== SISTEMA DE ARCHIVOS ========

B)ajar archivo S)ubir archivo
L)istar archivo N)ovedades
R)evisar archivo

AUXILIARES

V)olver al menú H)asta pronto
I)nformación A)yuda

¿Archivos>? V

Quedan 17 minutos.

========== MENU PRINCIPAL ==========
COMUNICACIONES

B)oletines A)yuda
C)orreo personal H)asta pronto
N)otas al Sysop P)ágina
O)perador U)tiles
S)istema archivos F)unciones

MENSAJES

D)ejar mensajes E)liminar msjs
L)eer mensajes M)irar mensajes
R)evisar mensajes

¿Menú Principal>? H

Son las 3:01 PM Tiempo: 1 Min y 25 Seg
Antes de dejar el sistema, por favor, conteste el siguiente cuestionario.

Gracias.

¿Tuvo problemas usando el BBS?(s/n)? n
Indique aquí sus sugerencias para aumentar las prestaciones del BBS. Todo en orden
Eso es todo.

Esperamos que vuelva a conectarse pronto.

¡Chau!
Chau, JORGE, gracias por llamar. Hasta pronto.

Naturalmente que esta conversación es supuesta, pero no diferirá mucho de la que desarrollará el usuario medio. Sin embargo, puede haber alguna diferencia en las pantallas pues este sistema está en proceso de "pulido" en cuanto a algunas opciones y pantallas.

Hugo Daniel Caro

GENERADOR DE MENUES

Tipo: Utilitario

La utilización de menúes para nuestros programas utilitarios es fundamental si el programa tiene que ser usado por algún operador inexperto.

Por eso un buen menú debe ser por sobre todo simple. Y si a esa simplicidad se le agrega una buena presentación, mucho mejor.

Contra la opinión general, agregar menúes a un programa no lo convierte a éste en algo demasiado extenso y, por el contrario, le aporta múltiples ventajas.

En el listado que ofrecemos, utilizamos un ejemplo típico como es el menú de una agenda (agregar, editar, buscar, grabar, fin).

En el mismo se recuadra una de las opciones y podemos subir o bajar ese cuadro con los cursores. Cuando nos decidimos por una de las opciones, apretamos la tecla RETURN y el programa finaliza diciéndonos cuál fue la opción que elegimos (variable CH de la línea 70).

Los lectores atentos notarán que la línea 60 ha sido anulada con un REM ('). En realidad ésta es la línea que se suele usar con los menúes en lugar de la 70, ya que de acuerdo al valor de CH el programa seguirá por uno u otro camino.

Otro truco destacable de programación es la utilización de una subrutina para generar el menú, una vez que le damos todos los datos. De esta manera para distintos menúes lo único que tendremos que hacer es colocar los ítems correspondientes y luego llamar a esa subrutina.

Creemos que este programa puede ser empleado por muchos de nuestros lectores como una rutina de gran utilidad: podrán mejorar así la presentación de sus programas.

Estructura del programa

10-30 Define pantalla. Presentación

40-70 Menú principal. Llamada a subrutina de generación de menú

80-170 Subrutina de generación de menú

200-210 Subrutina que devuelve el ASCII de la tecla presionada

Variables importantes

TI$=Nombre del menú

PO=Número de ítems que tiene el menú

PO$(PO)=Nombre de los ítems

CH=Opción elegida

```
10 CLEAR 1024:DEFINT A-Z
20 KEYOFF:SCREEN0:WIDTH39
30 DEFFNA$(X)=LEFT$("______________________________________________",X*2)
40 '-- Menu principal
50 TI$="Prueba de men#":PO$(1)="Agregar":PO$(2)="Editar":PO$(3)="Buscar":PO$(4)="Grabar":PO$(5)="Fin programa":PO=5:GOSUB90
60 'ONCHGOTO290,300,310,320,330
70 CLS:PRINT "Eligi":";CH;"= ";PO$(CH):END
80 '-- Generador men#
90 CLS:CH=1:M=0:FORI=1TOPO:IFLEN(PO$(I))>MTHENM=LEN(PO$(I))
100 NEXTI:L=(39-M)/2:LOCATE(39-LEN(TI$))/2,1:PRINTTI$:LOCATE,5:FORI=1TOPO:LOCATEL:PRINTPO$(I):PRINT:NEXTI:L=L-1:K$=FNA$(M):GOTO110
110 LOCATEL,2+CH*2:PRINT"┌";K$;"┐":LOCATEL:PRINT"│";:LOCATEL+M+1:PRINT"│":LOCATEL:PRINT"└";K$;"┘"
120 SOUND8,10:SOUND1,0:SOUND0,100:FORI=1TO50:NEXT:SOUND8,0
130 A$=INKEY$:IFA$=""THEN130ELSEX=ASC(A$)
140 LOCATEL,2+CH*2:PRINTSPC(M+2):LOCATEL,3+CH*2:PRINT" ":LOCATEL+M+1,3+CH*2:PRINT" ":LOCATEL:PRINTSPC(M+2)
150 IFX=31THENCH=CH+1:IFCH>POTHENCH=1
160 IFX=30THENCH=CH-1:IFCH=0THENCH=PO
170 IFX=13THENRETURNELSE110
200 '-- Devuelve el ASCII de tecla
210 A$=INPUT$(1):B=ASC(A$):RETURN
```

CONCURSO MENSUAL

PREMIO

MSX-TEST



Un soft a eleccion entre IDEA BASE, IDEA TEXT, o BASIC TUTOR.

CONCURSO MENSUAL

Auspiciado por TELEMATICA S.A. fabricante en Argentina de las computadoras personales TALENT MSX

Para participar en este certamen deben señalar cual es la información correcta que presenta cada ítem. Junto con las respuestas deben remitir los datos en el correspondiente cupón. Los premios se entregarán en la administración de la editorial. Los que no puedan concurrir pueden solicitar el envio pagando el franqueo contrareembolso. Los premios podrán ser reclamados dentro de los 120 días después de haber sido anunciados.

Ganador del MSX-TEST Nº 14

Alberto Martín Diel ·

Respuestas correctas

1) b.Vicecomodoro Marambio; 2) a.tres; 3) c.First Assembler; 4) c.Mozart; 5) c.18

MSX-TEST N. 18 - CIERRE 1 DE AGOSTO DE 1989

1. ¿Cuántos sprites se pueden crear y colocar en movimiento sobre la pantalla en todos los modos gráficos en la MSX1?

a. 8

b. 16

c. 32

2. ¿Cuántas tortugas puede mostrar en línea el MSX LOGO?

a. 2

b. 4

c. 8

3. ¿Qué significa la sigla PPI?

a. Port de entrada programable.

b. Protocolo de procesador independiente.

c. Interfase programable para periféricos.

4. El MSX-chart es un programa que hace...

a. ...gráficos de gestión.

b. ...carteles.

c. ...carátulas.

5. ¿Cuántos canales tiene la desgrabadora de Holter neurológico, aplicación médica de la MSX2?

a. 2

b. 4

c. 8

Nombre y apellido____________________

Dirección:____________________

Documento:__________ Edad:_____ Máquina:__________

Qué es lo que más me gusta de la revista:____________________

Que le agregaría:____________________

Que es lo que no me gusta:____________________

MSX2: VDP Y PANTALLA

Mostramos los distintos modos de trabajar con el procesador de pantalla de video, para que podamos sacarle provecho a las caracterísitcas gráficas de la MSX2.

En una nota anterior dijimos que la MSX2 cuenta con características gráficas muy especiales que se deben a su procesador de pantalla de video, o VDP.
Ahora continuamos desarrollando el uso de la V9938-VDP del MSX2 Turbo.
Ahora veremos como se puede acceder a la MSX-VIDEO directamente por los pórticos I/O sin utilizar el BIOS.

ACCESO A LOS REGISTROS

ESCRIBIENDO DATOS A LOS REGISTROS DE CONTROL
Los registros de control son write-only (escritura solamente).
Como describimos en una nota anterior, parte de los contenidos de los registros de control (R#0 R#23) se pueden obtener refiriéndose a la VDP(n) desde BASIC. Esta solamente lee el valor que se ha escrito en la work-area de RAM (F3DFH to F3E6H, FFE7H to FFF6H) utilizada para escribir los registros.
Existen tres formas diferentes, descriptas a continución, para escribir datos a los registros de control. Ya que la MSX accede a la MSX-VIDEO dentro de la rutina de interrupción de timer para examinar la ocurrencia de colisión de sprite, nótese que el procedimiento de acceso no inhibirá la interrupción si se accede a los registros en forma correcta como describiremos.

MSB	7	6	5	4	3	2	1	0	LSB
Port 1	D	A	T	O	S				1.Poner datos en port 1
Port 2	1	0	R5	R4	R3	R2	R1	R0	2.Luego poner número de registro con los bits más representativos
	Fijo en "10"		número de registro (0 a 46)						

Figura 1: Acceso directo a R#n

ACCESO DIRECTO

La primera forma es especificar directamente el dato y dónde va a ser escrito. La figura 1 ilustra el proceso. El

CENTROS DE ASISTENCIA AL USUARIO DE TALENT MSX

CAPITAL FEDERAL

Taller Logo de computación
Junín 1930

Centro de Computación Clínica
Asistencia al Usuario Discapacitado
Ramsay 2250 - Pabellón F
Tel. 784-2018

Barrio Norte
Uriburu 1063 - Tel. 83-6892/826-6692

Belgrano
Cabildo 735 - Tel. 772-9088

Flores
Gral. Artigas 354 - Tel. 612-3902

Palermo
Guatemala 4733 - Tel. 71-4124

San Telmo
Chile 1345 - Tel. 37-0051 al 54

GRAN BUENOS AIRES

Caseros
Medina y Alberdi - 1er piso - Tel. 750-8016.

Castelar
C. Casares 997 - Tel. 629-2247

Lanús
Caaguazú 2186 - Tel. 240-0378

Morón
Belgrano 160 - Tel. 629-3347

Ramos Mejía
Bolívar 55 - 1er. piso - Tel. 658-4777

San Justo
Perú 2515 (esq. Arieta)

Vicente López
Av. Maipú 625 - Tel. 797-672

Virreyes - Pdo. de San Fernando
Avellaneda 1697 - Tel. 745-7963

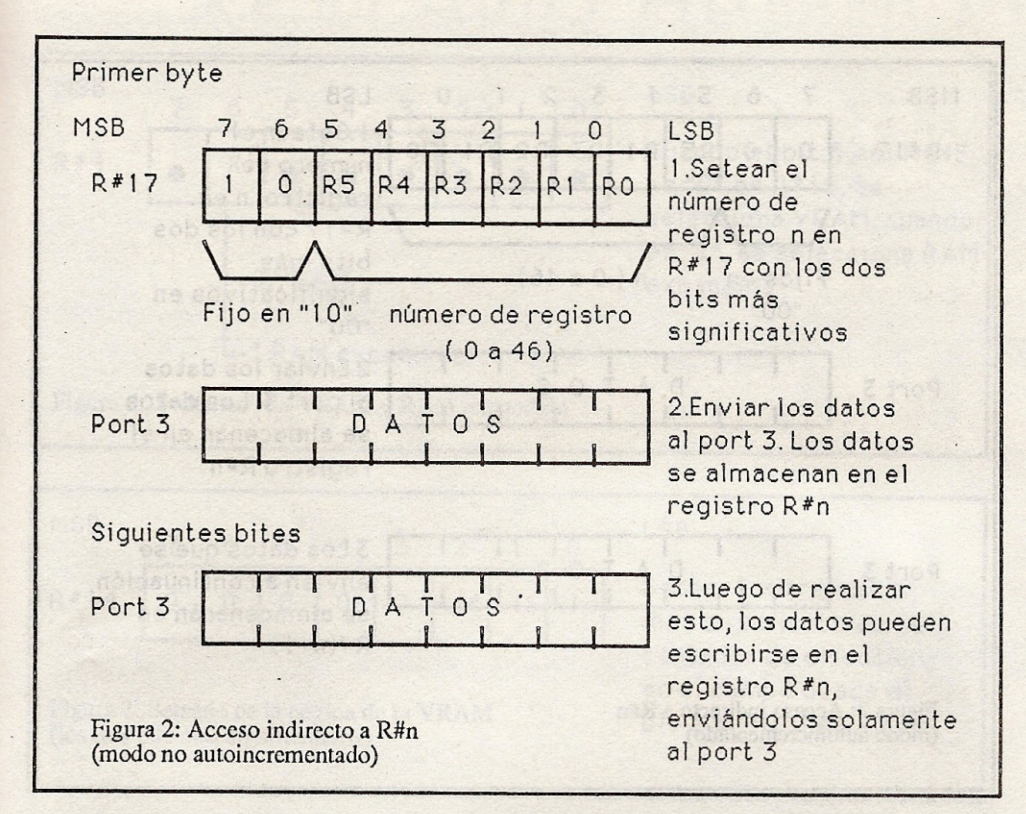


Figura 2: Acceso indirecto a R#n (modo no autoincrementado)

dato se escribe primero en el port N'1 y luego el número de registro destino en el mismo port, usando los cinco bits menos significativos. El bit más significativo se setea a 1 y el segundo bit se setea a 0. El valor sería 10XXXXXB en notación binaria, donde XXXXX es el número de registro destino.

También se usa el port N'1 para setear las direcciones de la VRAM, como describiremos más adelante. El bit más significativo del segundo byte enviado a este pórtico es el flag de dirección/registro que determinará la operación que tendrá lugar. Cuando el bit se setea a1, escribe datos al registro de control como vimos anteriormente.

ACCESO INDIRECTO
(modo no autoincrementado)

La segunda forma consiste en escribir datos al regitro especificado como el registro objetivo (R#17 contiene el puntero objetivo). Para empezar, hay que almacenar el número de registro al que se accederá en R#17 por acceso directo. El bit más significativo se setea a 1 y el segundo bit a 0. Luego, el valor sería 10XXXXXB en notación binaria, donde XXXXX es el número de registro objetivo. Después de haber hecho esto, el dato se puede escribir en el registro objetivo enviando datos al port N' 3. Este método se utiliza para enviar datos continuamente al mismo registro. Por ejemplo, para ejecutar comandos de VDP.

ACCESO INDIRECTO
(modo autoincrementado)

En la tercera forma se trata de escribir datos al registro indicado por R#17. Este se incrementa cada vez que enviamos datos al port N'3. Para empezar, hay que almacenar el número de registro inicial al que se accederá en R#17 por acceso directo. Los dos bits más significativos se setean a 0. Luego, el valor sería 00XXXXXB en notación binaria, donde XXXXX es el número de registro inicial. Ya que este método permite escribir datos en registros de control contiguos, es muy útil cuando muchos registros contiguos deben cambiarse en forma inme-

diata. Un ejemplo es el cambio del modo de pantalla.

SETEANDO UNA PALETA

Para setear datos en los registros de paleta en la MSX-VIDEO (P#0 to P#15), hay que especificar el número de registro de paleta en los cuatro bits menos significativos de R#16 (puntero de paleta de color) y enviar el dato al port N'2. Como los registros de paleta tienen una longitud de nueve bits, los datos deben ser enviados dos veces; el brillo rojo y azul primero, y luego el verde. El brillo está especificado en los tres bits menos significativos del segmento de cuatro bits. Véase la figura 4.

Después que el dato es enviado dos veces al port N'2, R#16 se incrementa automáticamente. Esta característica facilita la inicialización de todas las paletas.

LEYENDO REGISTROS DE ESTADO

Los registros de estado son read-only (lectura solamente). Su contenido se puede leer desde el port N'1 seteando el número de registro de estado en los cuatro bits menos significativos de R#5 (puntero de registro de estado), como se ve en la figura 5. Los cuatro bits más significativos se setean a 0. Luego, el valor sería 0000XXXXB en notación binaria, donde XXXX es el número de registro de estado. Se deben inhibir las interrupciones antes de acceder al registro de estado. Después de completar la tarea deseada, R#15 debe ser seteado a 0 y restablecer las interrupciones.

ACCESO A LA VRAM DESDE EL CPU

Cuando se debe acceder a una direc-

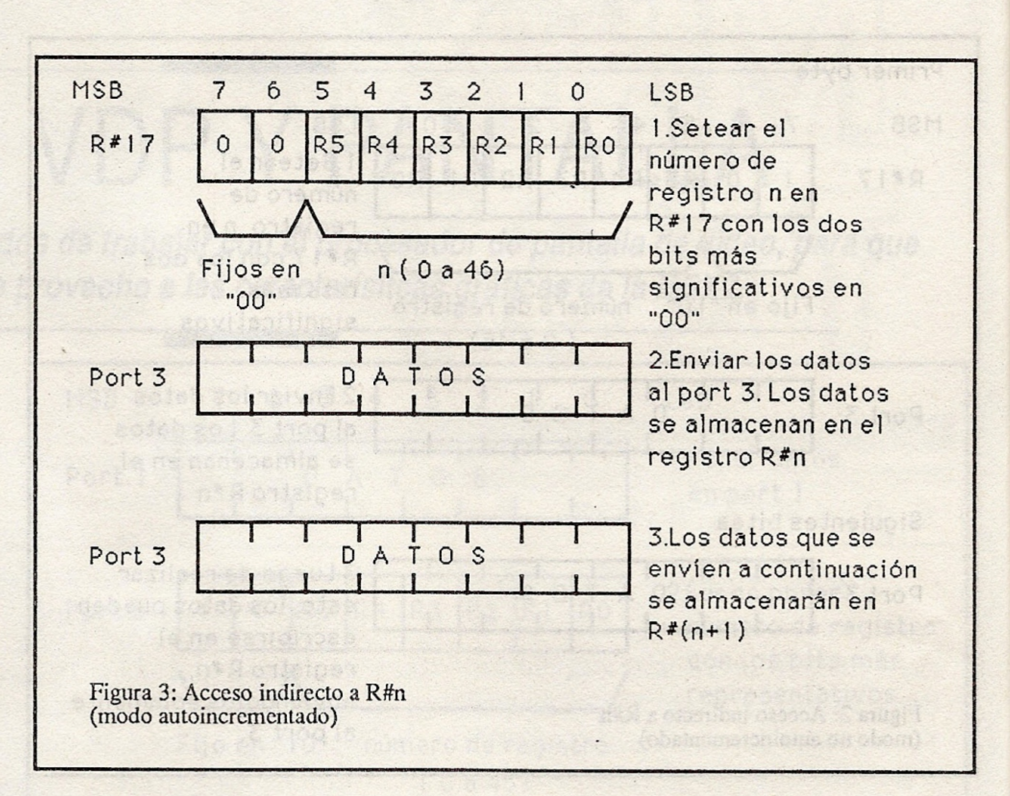


Figura 3: Acceso indirecto a R#n (modo autoincrementado)

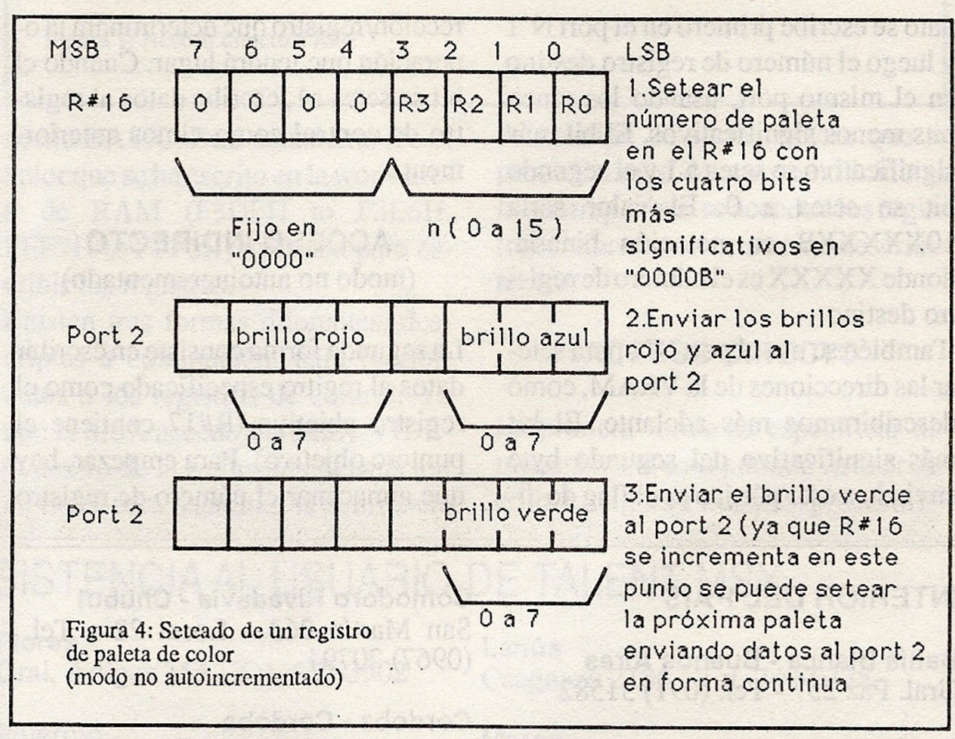


Figura 4: Seteado de un registro de paleta de color (modo no autoincrementado)

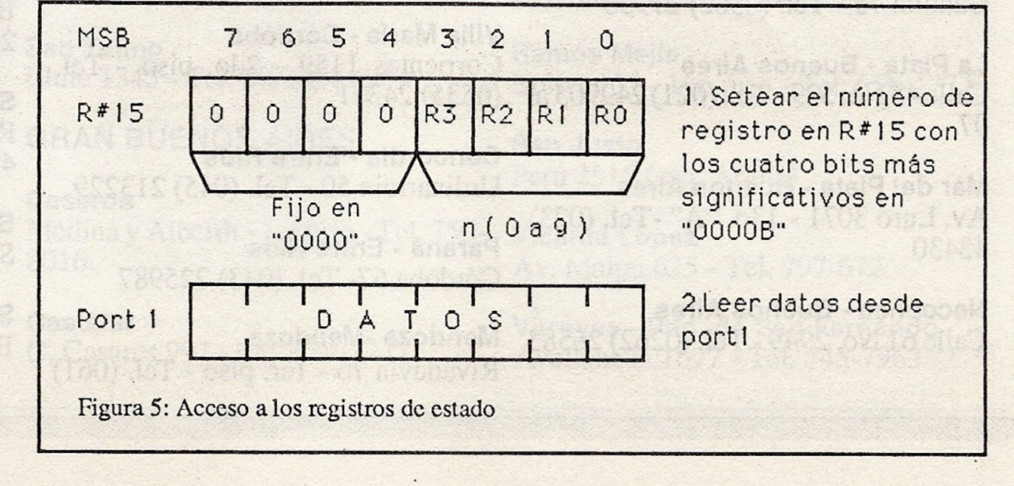


Figura 5: Acceso a los registros de estado

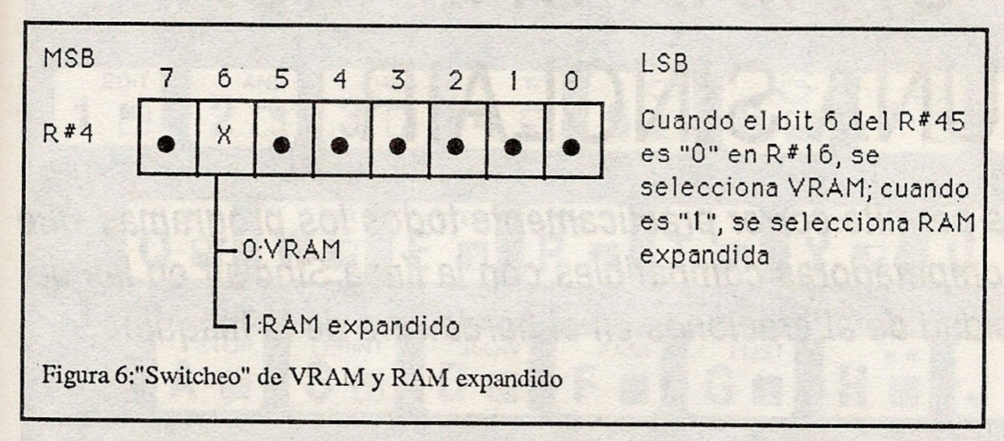


Figura 6:"Switcheo" de VRAM y RAM expandido

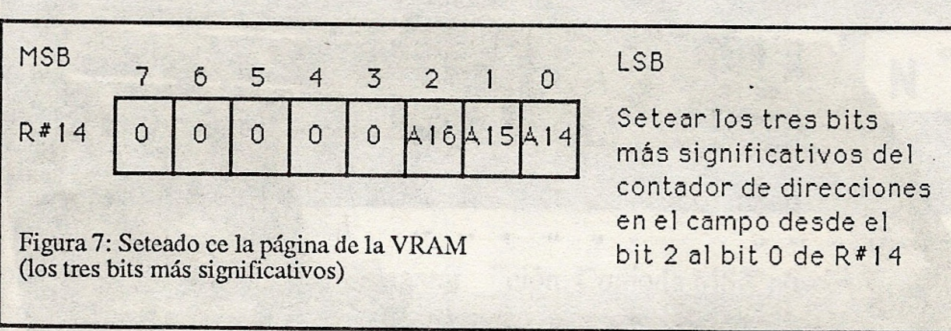


Figura 7: Seteado ce la página de la VRAM (los tres bits más significativos)

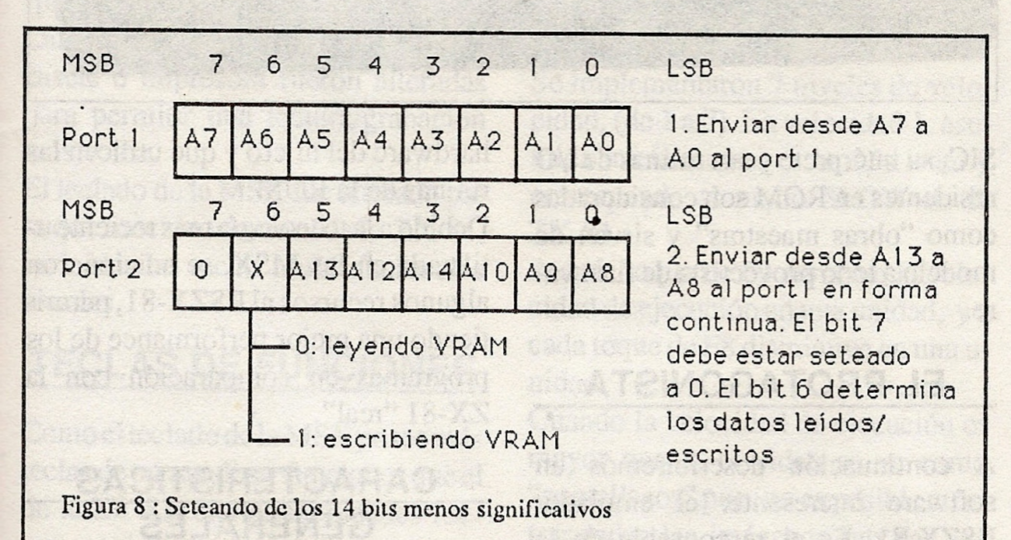


Figura 8 : Seteando de los 14 bits menos significativos

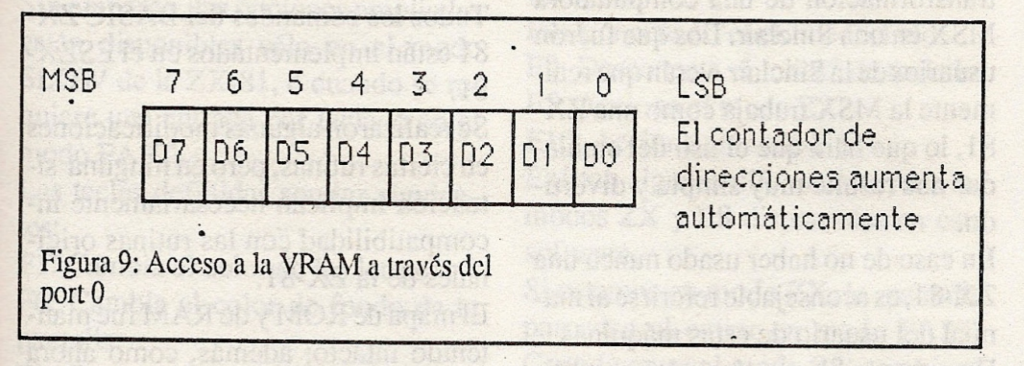


Figura 9: Acceso a la VRAM a través del port 0

ción de VRAM desde el CPU, se deben cumplir los siguientes pasos.

CONMUTAR BANCOS

Los primeros 64K bytes de VRAM (00000H a 0FFFFH) y los 64K bytes de RAM expandida residen en el mismo espacio de direccionamiento, tal como lo ve la MSX-VIDEO.

La conmutación de bancos es utilizada para que ambas puedan estar en línea al mismo tiempo. Como la MSX2 no usa RAM expandida, siempre hay que seleccionar el banco de VRAM. El bit 6 de R#45 controla la conmutación de bancos.

SETEADO DE LA PAGINA DE LA VRAM (los tres bits más significativos).

La dirección de 17 bits para acceder a los 128K de VRAM se setea en el contador direccional (A16 to A0) R#14 los tres bits más significativos (A16 to A14). Luego este registro se puede ver "switcheando" entre ocho páginas de 16K bytes de VRAM.

SETEADO DE LA DIRECCION DE LA VRAM (los 14 bits menos significativos).

Los 14 bits menos significativos de la dirección deben ser enviados al port N'1 en dos bytes, como se verá en la figura 8. Al enviar el más significativo bit del segundo byte, asegurémonos. Este setea el flag de dirección/registro a modo direccional. El segundo bit más significativo setea el flag de lectura/escritura. 1 significa escribir a VRAM y 2 significa leer de VRAM.

LEYENDO Y ESCRIBIENDO DE LA VRAM

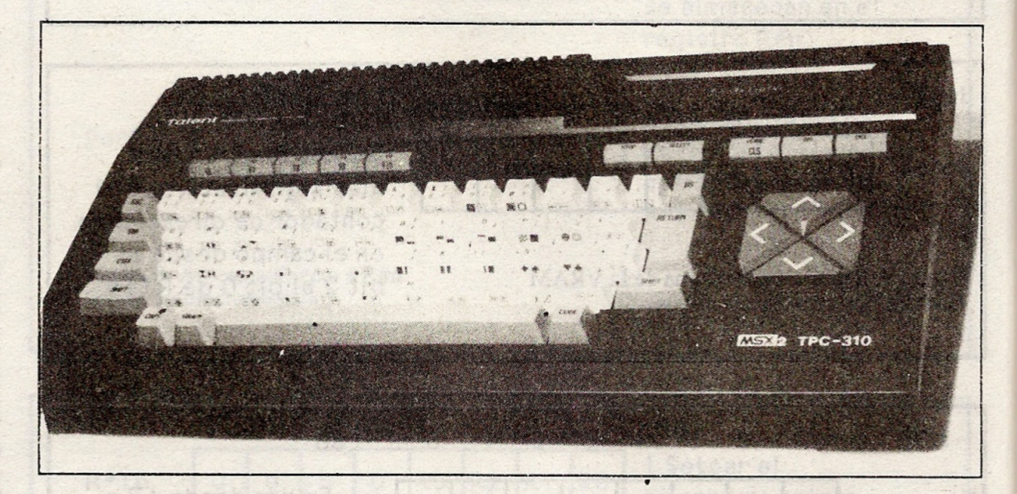
Después de seteado el valor en el contador de direcciones, se pueden leer o escribir datos a través del port N'0. El flag de lectura/escritura se setea al mismo tiempo que A13 a A8 del contador de direcciones. Cada vez que un byte de datos se lea o escriba al port N'1, el contador de direcciones se incrementará automáticamente. Esta característica permite acceder fácilmente a memoria contigua en VRAM.

Bibliografía: MSX2 Technical Handbook

Jorge Agnese - Hugo D. Caro

TRANSFORMAR UNA MSX EN UNA SINCLAIR

El software que presentamos permite correr prácticamente todos los programas que fueron desarrollados para computadoras compatibles con la línea Sinclair en las de la norma MSX sin necesidad de alteraciones en el hardaware de la máquina



Existen varias formas de implementar un emulador, pero básicamente depende de hacia dónde se apunte. La variante difícil es mediante el hardware.

Cuando se desea emular una computadora (generalmente más pequeña que la "simuladora"), se arma una plaqueta para insertar en la computadora "simuladora" con la CPU, RAM y demás yerbas, de manera que prácticamente del "host" sólo se usa la pantalla y el teclado.

Otra forma de emulación es por software. En este caso, la dificultad con que se tropieza es la lentitud de ejecución, pues se deben interpretar los comandos de una CPU diferente al "host".

Sin embargo, las MSX poseen "parientes" de CPU: la línea Sinclair.

Las microcomputadoras de la línea ZX-81 fueron las primeras en lograr difusión a nivel mundial como computadoras hogareñas. Fue cuando Sir Clive Sinclair logró producir una microcomputadora realmente revolucionaria, la Sinclair ZX-81. En su época fue la única microcomputadora en el mundo cuyo costo era inferior a los U$S 300. En nuestro país la comercializó Czerweny bajo la sigla CZ-1000, CZ-1500 y Arvoc, con la versión brasileña TK-83/85.

Los que se iniciaron en la ZX-81 sienten orgullo de ser llamados "sinclairistas" y no admiten que se hable mal de la ZX-81. La estructura de su BASIC, su intérprete y sus rutinas de I/O residentes en ROM son consideradas como "obras maestras" y sirven de modelo a todo proyectista de firmware.

EL PROTAGONISTA

A continuación describiremos un software interesante: el emulador ESZX-81. Es el responsable de la transformación de una computadora MSX en una Sinclair. Los que fueron usuarios de la Sinclair, verán que realmente la MSX trabaja como una ZX-81, lo que hará que el uso del emulador nos resulte muy simple y divertido.

En caso de no haber usado nunca la ZX-81, es aconsejable referirse al manual del usuario de estas máquinas.

Esto es posible gracias a la versatilidad y simplicidad de la línea MSX y también a la excelente estructura de entrada/salida de la línea Sinclair.

Asimismo, los programas escritos en lenguaje de máquina correrán en el ESZX-81, siempre que no accedan al hardware del micro y que utilicen las rutinas de la ROM.

Debido a la tecnología más reciente utilizada en las MSX, se adicionaron algunos recursos al ESZX-81, permitiendo una mejor performance de los programas en comparación con la ZX-81 "real".

CARACTERISTICAS GENERALES

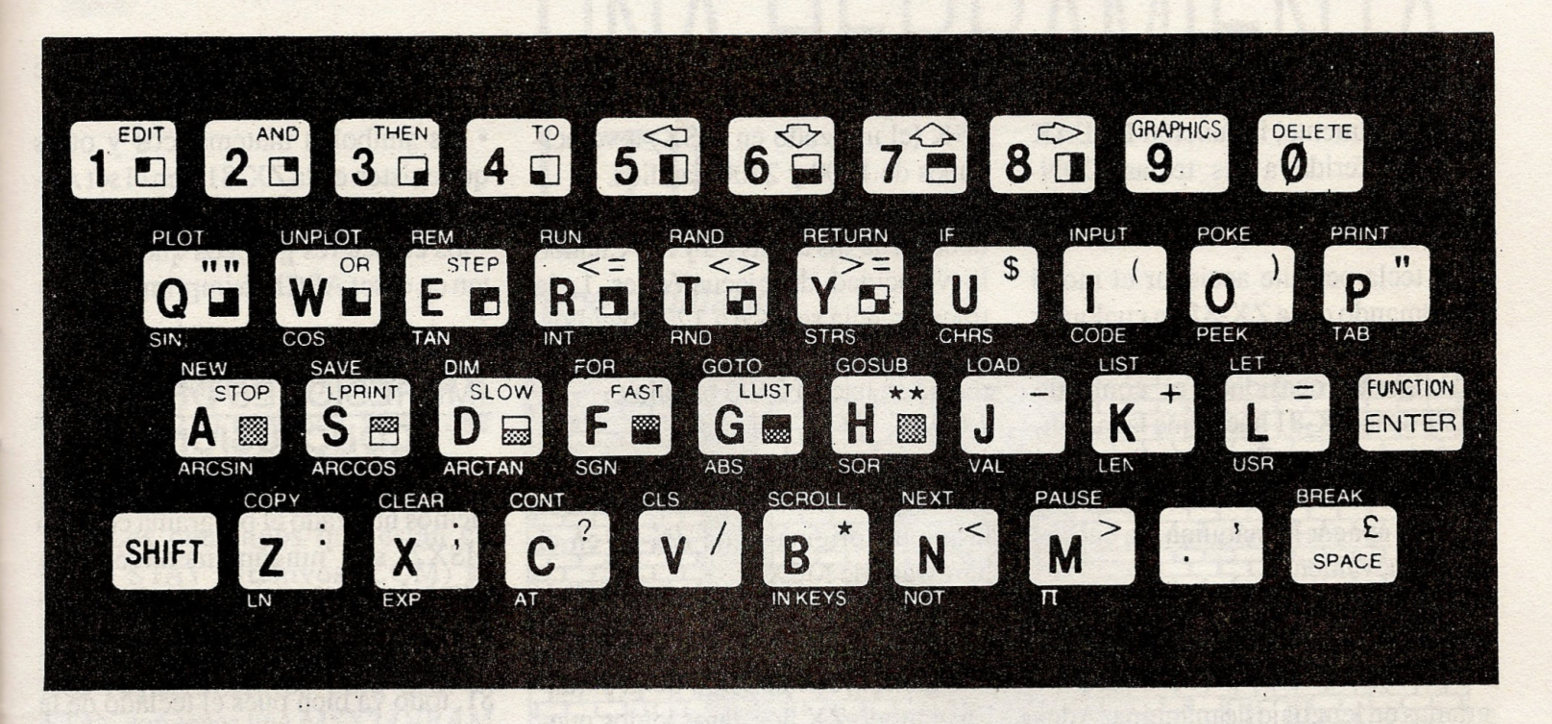
Todos los comandos del BASIC ZX-81 están implementados en el ESZX-81.

Se realizaron algunas modificaciones en ciertas rutinas, pero en ninguna situación implican necesariamente incompatibilidad con las rutinas originales de la ZX-81.

El mapa de ROM y de RAM fue mantenido intacto; además, como ahora tanto la ROM como la RAM de la ZX-81 están instaladas en RAM, es posible alterar la ROM a gusto (aunque no es muy saludable si no se tiene conocimiento de causa...).

Esto significa, por ejemplo, que los caracteres de la ZX-81 pueden ser redefinidos a través del BASIC y se

Figura 1



pueden, de esta manera, implementar programas de alta resolución gráfica (para la ZX-81) fácilmente.

Las rutinas de entrada y salida para casete e impresora fueron alteradas para permitir una lectura/grabación más veloz.

El teclado de la MSX fue alterado para que funcione como el de la ZX-81, incluidas las teclas multifuncionales usuales.

TECLAS DE FUNCIONES

Como el teclado de la MSX posee más teclas de comandos y funciones que el de la ZX-81, ciertas facilidades fueron asociadas a ellas.

Sin embargo, las opciones ampliadas están disponibles sólo en el modo SLOW de la ZX-81, o cuando se requiere una entrada por teclado en el modo FAST.

Las teclas definidas son las siguientes:

F1: Cambia el color de las letras.
F6: Cambia el color de fondo de la pantalla.
F2: Conecta el motor del grabador (equivale a 'MOTOR ON' del MSX-BASIC).
F7: Desconecta el motor del grabador ('MOTOR OFF' del MSX-BASIC).
F3: Aumenta la velocidad de ejecución.
F8: Disminuye la velocidad de ejecución. Como la MSX posee un procesador dedicado al video (la VDP), el ESZX-81 puede ejecutar programas más rápidamente que la ZX-81.

Se implementaron 7 niveles de velocidad, (de 1 a 7). La velocidad 1, asumida inicialmente, es la más lenta y corresponde al modo SLOW de la ZX-81.

A cada toque de F3 aumenta la velocidad de ejecución en una unidad, y a cada toque de F8 disminuye en una unidad.

Cuando la velocidad de ejecución es mayor que 2, el video se ve como "pantallazos" pues no es posible mostrar todas las imágenes (no dan los tiempos).

F4: Conecta el 'click' de teclado.
F9: Desconecta el 'click' de teclado.
F5: Acciona el modo ZX.
F10: Acciona el modo ES.

Existen algunas diferencias entre los modos ZX y ES disponibles en este software.

Si estamos en modo ZX, la pantalla parpadea al entrar en modo FAST. Cuando entra al modo ES, cambia el color de borde de la pantalla y ocurren algunos cambios operativos en el ESZX-81.

El primer cambio es que los códigos de error (que en la ZX-81 están indicados por códigos numéricos), se muestran con mensajes de error completos.

Además, la pantalla ya no parpadea en el modo FAST y está siempre actualizada.

Por último, los comandos LOAD y SAVE se modificaron de acuerdo al modo, como se explicará posteriormente.

TECLAS ESPECIALES

Las teclas descriptas a continuación, del mismo modo que las de función, sólo tienen efecto en el modo SLOW o durante una entrada vía teclado en el modo FAST.

STOP

La tecla STOP funciona como en las MSX, 'congelando' la ejecución de un programa BASIC y, además, en este caso detiene programas en código de máquina.

Si se está ejecutando el comando LOAD en modo SLOW, STOP lo interrumpirá definitivamente.

SHIFT+STOP

Presionando simultáneamente estas dos teclas, se continúa con la ejecución del programa interrumpida con STOP.

CONTROL+STOP

La tecla BREAK de la ZX-81 estaba en la misma tecla que SPACE, que era muy chica y difícil de accionar.

Como en las MSX la tecla de espacio es más grande y puede ser accionada accidentalmente, la función BREAK fue transferida a las teclas CONTROL+STOP.

HOME

Esta tecla permite accionar el modo de comando de la ZX-81 en cualquier momento.

En algunas ocasiones el comando LIST de la ZX-81 hacía que la máquina entrara en un 'loop' de listado, y cuando esto ocurría estábamos obligados a apagar la máquina y a encenderla nuevamente.

Ahora, presionando HOME, el ESZX-81 volverá al modo comando, aun cuando estuviera corriendo programas en lenguaje de máquina. Además, la dirección inicial de la tabla de caracteres es restablecida en&HE100.

ESC

Esta tecla equivale a FUNCTION de la ZX-81, (SHIFT+NEW LINE).

TAB

Equivalente a GRAPHICS, (SHIFT 9 de la ZX-81).

INSERT

Equivalente a EDIT, (SHIFT 1).

DELETE

Equivalente a RUBOUT, (SHIFT 0).

BS

Equivalente a RUBOUT, (SHIFT 0).

TECLAS DE CURSOR

Fueron asociadas a las teclas SHIFT+5, SHIFT+6, SHIFT+7 y SHIFT+8 de la ZX-81, y mueven, asimismo, el cursor.

JOYSTICK

EL joystick de MSX funciona como en la ZX-81; sus direcciones corresponden a las teclas 5,6,7 y 8. Ambos disparadores corresponden a la tecla 0.

LOAD Y SAVE

Las rutinas de LOAD y SAVE del ESZX-81 funcionan de forma diferente en los modos ZX y ES.

En el modo ZX, funcionan como en el ZX-81, esto significa que la velocidad de transferencia es de 300 baudios en norma ZX-81.

En el modo ES, tanto la lectura como la grabación se relizan en el sistema FSK (el utilizado en MSX) a velocidades de 1200 y 2400 baudios.

Para seleccionar las velocidades se deben usar las teclas F3 y F7. Cuando la velocidad de ejecución es 1, la transferencia se hace a 1200 baudios. Si la velocidad es mayor, la transferencia se hace a 2400 baudios.

LOAD

La rutina del comando LOAD fue alterada internamente para poder leer las cintas originales de ZX-81 en el hardware de MSX.

Al ejecutar el comando LOAD, se exhibe en pantalla el nombre del programa que está siendo leído de la cinta. En el modo ZX, los datos leídos pueden ser monitoreados por el audio del televisor o monitor.

Como aclaración final, hay que destacar que desde el modo ZX se deben cargar cintas grabadas en modo ZX únicamente; lo mismo vale para el modo ES.

SAVE

El comando SAVE, al ser ejecutado, muestra el mensaje "aguarde" en pantalla, al tiempo que graba el programa en cinta.

Lo que se está grabando también puede ser monitoreado por el parlante del televisor, desde el modo ZX.

Tanto el comando LOAD como el comando SAVE pueden ser cancelados presionando la tecla STOP en vez de CONTROL+STOP, desde el modo ZX.

COMANDOS DE IMPRESION

Los comandos de impresión, LLIST, LPRINT y COPY, fueron modificados internamente para utilizar impresoras con el set de caracteres ASCII. De esta forma al usar uno de los comandos mencionados el resultado será el siguiente:

- las letras en video normal se imprimen como mayúsculas;
- las letras en video inverso se imprimen como minúsculas;
- los números del 0 al 9, tanto en video normal como en video inverso, se imprimen normalmente;
- los símbolos matemáticos y otros que existen en la ZX-81 y en el set ASCII se imprimen normalmente;
- los caracteres gráficos que no existen en el set ASCIIse imprimen como ?'.

IMPRESIONES DE USO Y CONCLUSIONES

Hemos utilizado el programa con una MSX2 sin ningún inconveniente (también corre perfectamente en una MSX1).

Si se tiene cerca un manual de la ZX-81, todo va bien pues el teclado de la misma es un tanto complicado.

Para solucionar este problema, en la Figura 1 transcribimos el teclado original de la ZX-81 para que pueda comparar el usuario MSX no habituado al mismo.

En mi caso particular, por el año 1983 tuve la posibilidad de dar un curso de BASIC sobre una de estas máquinas. Eran tiempos duros en los que el "sueño" de una máquina por alumno era no más que eso: un sueño lejano.

Sin embargo, los resultados obtenidos fueron muy buenos, considerando las limitaciones (prácticamente era un curso teórico...). Cuando los alumnos se sentaban frente a la ZX-81, se veía claramente que esta computadora les resultaba bastante fácil de usar y permitía lograr los objetivos (algo limitados) del curso: aprender BASIC.

Cuando tuve en mis manos el ESZX-81, lo primero que hice fue desempolvar los viejos casetes con programas de la Sinclair. El resultado fue asombroso: de 8 casetes con 4 o 5 juegos o programas cada uno, no funcionaron sólo aquellos que, usando el código de máquina, accedían directamente al hardware. Los programas en código de máquina "prolijos" (que usan la ROM) funcionaron perfectamente.

Este emulador es un programa de alta calidad.

Hugo Daniel Caro

UNA HERRAMIENTA UTIL

Tipo: Utilitario

Este programa instala en la MSX2 el comando STAY que simula una RAM-DISK usando la VRAM Y el MSDOS.

Los comandos que están en modo STAY (archivos .COM) se cargan desde la RAM de video en lugar de ser buscados en el disco.

Los requerimientos para usar el STAY son tener una MSX2 (VRAM 128K), el MSXDOS. SYS versión 1.03 de fecha 23/8/85 y el COMMAND.COM versión 1.11 del 2/9/85.

COMANDOS

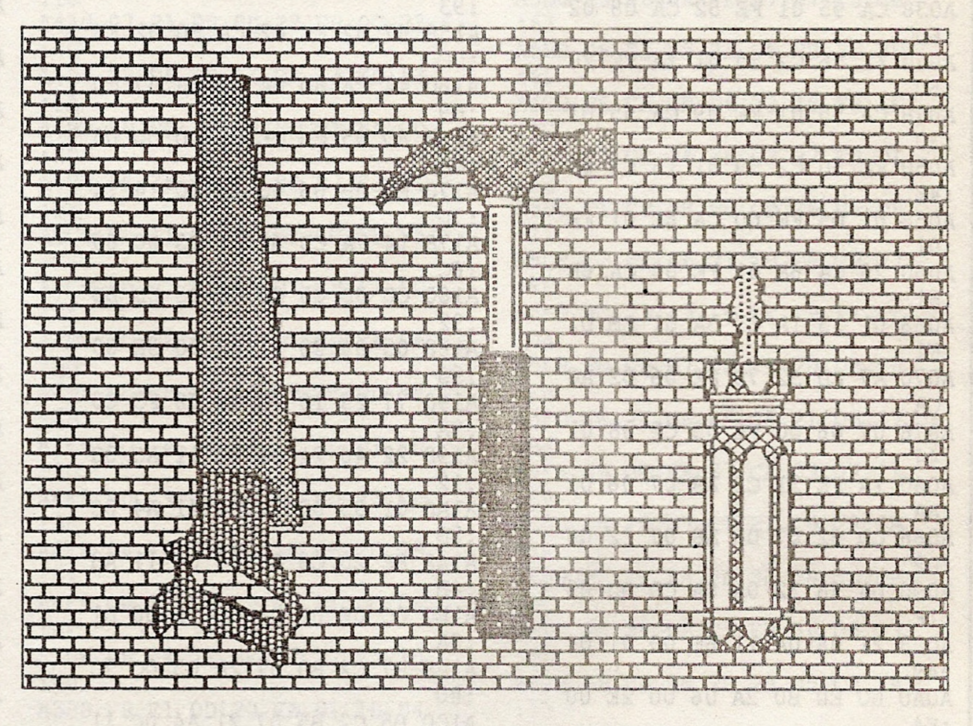
Hay varios comandos que tenemos disponibles cuando ingresamos en modo STAY. Ellos son:

STAY (a secas) muestra una ayuda (help)

STAY/D lista el directorio

STAY/R es un release (recupera el estado anterior del COMMAND.COM)

STAY/B vuelve al BASIC



FORMA DE CARGAR EL PROGRAMA

Primero se debe copiar el cargador. Con él ingresamos todos los datos del listado 1.

Una vez concluida esta tarea grabamos esos datos con BSAVE "stay.obj",&HA000,&HABFF

Luego copiemos, grabemos y hagamos correr el listado 2.

Cuando se va al BASIC o se apaga la máquina se lo debe hacer con STAY/B. Si no se hace así queda el COMMAND.COM modificado.

Cargador

```
100 SCREEN 0:CLEAR 200,&H87FF:
DEFINTA-Z:Z$="0000"
110 ON ERROR GOTO 300
120 PRINT:PRINT "*";:GOSUB
260:PRINT A$
130 IF A$="M" THEN 150
135 IF A$="Q" THEN 310
140 IF A$="D" THEN 210 ELSE
PRINT:GOTO 120
150 LINE INPUT A$:A=VAL("&H"+ A
$)
160 PRINT:GOSUB 280:V=PEEK(A):
GOSUB 290:PRINT"-";
170 GOSUB 240:L=V*16:IF E=1THEN
190 ELSE GOSUB 240:L=L+V:IF E=1
THEN 190 ELSE POKE A,L:A=A+1
180 GOTO 160
190 IF A$=CHR$(8) THENA=A-1 EL
SE  IF A$=CHR$(32) THEN A=A+1 E
LSE IF A$=CHR$(13) THEN 120
200 GOTO 160
210 LINE INPUT A$:A=VAL("&H "+
A$)
220 FOR L=0 TO 15:GOSUB 280:FOR
M=0 TO 7:V=PEEK(A):S=S+V:GOSUB
290:A=A+1:NEXT:PRINT":";:V=S:GO
SUB 290:PRINT: NEXT
230 PRINT:GOSUB 260:IFA$<>" " T
HEN 120 ELSE 220
240 E=0:GOSUB 260:IF A$<CHR$(48
) THEN E=1:RETURN ELSE IF A$>CH
R$(70) THEN 240 ELSE IF A$>CHR$
(57) AND A$<CHR$(65) THEN 240
250 V=VAL("&H"+A$):PRINT A$;:RE
TURN
260 A$=INKEY$:IF A$="" THEN 260
ELSE IF A$>CHR$(96) AND A$<CHR$
(123) THEN A$=CHR$
(ASC(A$)-32)
270 RETURN
280 A$=HEX$(A):PRINT LEFT$(Z$,4
-LEN(A$))+A$+" ";:S=PEK(VARPTR(
A))+PEEK(VARPTR(A)+1):RETURN
290 A$=RIGHT$(HEX$(V),2):PRINT
LEFT$(Z$,2-LEN(A$))+A$+" ";: RE
TURN
300 RESUME NEXT
310 ON ERROR GOTO 0:END
```

Listado 1

```
A000 11 22 0A 0E 09 CD 05 00
:0F
A008 3A C1 FC 21 2D 00 CD 0C
:07
A010 00 FE 01 C2 A3 07 11 81
:E6
A018 00 CD 2E 07 CD 83 05 CA
```

```
:0A
A020 11 02 11 81 00 1A 13 FE
:B9
A028 0D CA 95 01 FE 2F CA 34
:81
A030 01 C3 25 01 1A 13 FE 0D
:0B
A038 CA 95 01 FE 52 CA 08 02
:6D
A040 FE 56 C2 25 01 1A FE 20
:5D
A048 CA 55 01 FE 09 CA 55 01
:30
A050 FE 30 C2 59 01 13 C3 45
:4E
A058 01 D5 FE 0D CA 6E 01 FE
:01
A060 20 CA 6E 01 FE 09 CA 6E
:81
A068 01 13 1A C3 5A 01 EB D1
:F1
A070 A7 ED 52 7D FE 06 D2 A8
:CA
A078 07 06 00 FE 05 C2 88 01
:44
A080 1A FE 31 C2 A8 07 06 01
:AA
A088 CD 62 07 DA A8 07 22 08
:D2
A090 0A 78 32 0A 0A CD 9D 03
:1E
A098 21 08 0A 11 6E 09 01 04
:A9
A0A0 00 ED B0 2A 06 00 2E 00
:E4
A0A8 25 25 25 F9 EB 14 21 F4
:65
A0B0 07 ED 53 F4 09 01 00 02
:30
A0B8 ED B0 CD 8D 05 21 81 00
:87
A0C0 11 A3 0C CD 41 06 C2 9A
:19
A0C8 07 D5 E5 CD 9A 02 E1 D1
:C5
A0D0 DA D9 01 CD 41 06 CA C9
:44
A0D8 01 CD 0C 05 2A 06 00 2E
:26
A0E0 00 E5 D1 15 15 15 01 00
:DF
A0E8 01 ED B0 3A 07 00 3D 3D
:42
A0F0 3D ED 5B 06 00 1E 00 CD
:5F
A0F8 61 04 2A 06 00 25 25 25
:ED
A100 2E 00 CD 87 04 C3 00 00
:32
A108 CD 8D 05 CD CE 05 C3 00
:AB
A110 00 2A 06 00 24 2E 00 22
:8D
A118 F4 09 21 81 00 11 A3 0C
:48
A120 CD 41 06 C2 44 02 D5 E5
:BF
A128 CD 9A 02 E1 D1 DA 36 02
:16
A130 CD 41 06 CA 26 02 CD 0C
:C8
A138 05 2A 06 00 2E 00 CD 87
:A0
A140 04 C3 00 00 11 81 00 CD
:0F
A148 4E 07 CA 9A 07 1A FE 2F
:F0
A150 C2 9A 07 13 1A FE 52 CA
:93
A158 67 02 FE 42 CA 80 02 FE
:DC
A160 44 CA 8D 04 C3 9A 07 CD
:B9
A168 8D 05 CD CE 05 CD DD 05
:CA
A170 ED 5B 06 00 1E 00 14 14
:7D
A178 14 7A CD 61 04 C3 00 00
:6C
A180 CD 8D 05 CD CE 05 CD DD
:92
A188 05 06 80 21 80 00 3E 00
:53
A190 77 23 10 FC CD 68 F3 C3
:7A
A198 22 40 21 D1 0C 11 AC 0C
:12
A1A0 01 03 00 ED B0 21 A4 0C
:5B
A1A8 FE 20 C8 CD F7 04 11 A3
:4B
A1B0 0C 0E 0F D5 CD 05 00 D1
:8A
A1B8 B7 C2 B0 07 2A B5 0C 7C
:80
A1C0 B5 C2 B5 07 21 A4 0C 11
:FE
A1C8 F8 09 01 08 00 ED B0 2A
:BA
A1D0 F4 09 24 2E 7A 5E 23 56
:89
A1D8 23 4E 23 46 EB 78 32 0B
:63
A1E0 0A 22 05 0A 79 32 07 0A
:E0
A1E8 ED 4B B3 0C ED 43 03 0A
:1D
A1F0 57 AF 09 17 82 5F FE 02
:F0
A1F8 20 05 7C B5 C2 E7 07 22
:11
A200 08 0A 7B 32 0A 0A 3A 0B
:01
A208 0A 3C FE 09 D2 EC 07 32
:2D
A210 0B 0A 11 ED 0C 0E 1A CD
:FD
A218 05 00 11 A3 0C 21 00 00
:CF
A220 22 C4 0C 22 C6 0C 23 22
:14
A228 B1 0C 2A B3 0C 0E 27 CD
:91
A230 05 00 B7 C2 BA 07 CD B5
:AA
A238 03 2A F4 09 24 2E 7D 46
:28
A240 21 01 1B 05 04 CA 4E 03
:4A
A248 2C 2C 2C 2C 10 FA 22 AC
:70
A250 0C 11 A4 0C CD 32 05 C2
:7B
A258 00 00 2A F4 09 24 2E 7D
:DE
A260 46 2E 80 05 04 CA 6E 03
:20
A268 11 10 00 19 10 FA EB 21
:38
A270 F8 09 01 10 00 ED B0 21
:B8
A278 08 0A ED 5B F4 09 14 1E
:71
A280 7A 01 04 00 ED B0 21 03
:28
A288 0A 5E 23 56 EB 11 00 00
:C5
A290 CD EA 05 11 9C 0B 0E 09
:73
A298 CD 05 00 A7 C9 3A C1 FC
:21
A2A0 21 06 00 F5 E5 CD 0C 00
:C2
A2A8 32 72 09 E1 F1 2C CD 0C
:6C
A2B0 00 32 73 09 C9 2A F4 09
:86
A2B8 24 2E 7A 5E 23 56 23 7E
:2C
A2C0 21 ED 0C ED 4B B3 0C F5
:EE
A2C8 C5 D5 E5 CB 38 CB 19 11
:5F
A2D0 00 00 7E 23 83 5F 7E 23
:0C
A2D8 8A 57 0B 78 B1 20 F3 ED
:FD
A2E0 53 01 0A E1 D1 C1 F1 C5
:6F
A2E8 F5 E5 D5 2A F4 09 24 2E
:10
A2F0 7F 7E 3C 4F 3A B0 F3 FE
:4B
A2F8 29 D2 1A 04 C5 11 10 04
:EB
A300 3A E0 F3 E6 A7 B3 F3 ED
:15
A308 79 3E 81 ED 79 3A DF F3
:92
A310 E6 F1 B2 ED 79 3E 80 ED
:82
A318 79 C1 D1 E1 F1 1F 7A 17
:75
A320 17 17 E6 07 ED 79 32 ED
:88
A328 FF 3E 8E ED 79 7B ED 79
:FA
A330 7A E6 3F F6 40 ED 79 D1
:F4
A338 E5 2A F4 09 24 2E 7F 7E
:43
A340 E1 4F 7E ED 79 23 1B 7A
:B4
A348 B3 20 F7 0C 3A DF F3 F3
:BD
A350 ED 79 3E 80 ED 79 3A E0
:8C
A358 F3 ED 79 3E 81 ED 79 FB
:61
A360 C9 32 24 F3 32 26 F3 21
:66
A368 10 0A 4E 23 46 23 F5 78
:49
```

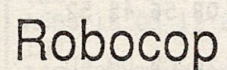
```
A370 B1 CA 7E 04 F1 D5 E5 EB
:7B
A378 09 77 E1 D1 18 EC 4E 23
:8F
A380 46 EB 09 F1 3D 77 C9 2E
:BE
A388 17 7E 3C 77 C9 11 7A 0A
:8E
A390 0E 09 CD 05 00 2A 06 00
:01
A398 24 24 2E 80 E5 2A F4 09
:EA
A3A0 24 2E 7D 7E E1 F5 F5 CD
:CD
A3A8 F7 04 5E 23 56 E5 EB 11
:9B
A3B0 00 00 CD EA 05 E1 3E 0D
:D0
A3B8 CD 32 06 3E 0A CD 32 06
:3A
A3C0 11 04 00 19 F1 3D B7 20
:1B
A3C8 DD 11 29 0C 0E 09 CD 05
:F4
A3D0 00 F1 F5 C6 30 CD 32 06
:C9
A3D8 11 2E 0C 0E 09 CD 05 00
:1C
A3E0 D1 15 3E 20 CA E9 04 3E
:21
A3E8 73 CD 32 06 3E 20 CD 32
:BD
A3F0 06 CD 0C 05 C3 00 00 06
:95
A3F8 08 7E CD 32 06 23 10 F9
:9F
A400 11 34 0C E5 CD DF 07 E1
:B2
A408 23 23 23 C9 2A F4 09 24
:65
A410 2E 7A 5E 23 56 23 4E 21
:F9
A418 00 00 3E 02 91 A7 ED 52
:9F
A420 06 00 CB 10 90 5F 16 00
:CE
A428 CD EA 05 11 85 0A CD DF
:F0
A430 07 C9 0E 1A CD 05 00 11
:C3
A438 C8 0C 0E 0F CD 05 00 B7
:62
A440 C2 CC 07 21 00 00 22 EB
:AB
A448 0C 23 22 D6 0C 2A F4 09
:42
A450 24 2E 7D 46 21 93 0C 05
:C2
A458 04 CA 60 05 23 23 10 FC
:6D
A460 5E 23 56 ED 53 E9 0C 11
:05
A468 C8 0C 21 0A 00 0E 26 CD
:E8
A470 05 00 B7 C2 D1 07 11 C8
:17
A478 0C 0E 10 CD 05 00 B7 C2
:5D
A480 D6 07 C9 2A 06 00 24 2E
:10
A488 00 7E FE 53 C9 11 39 0C
:D6
A490 0E 1A CD 05 00 11 C8 0C
:C7
A498 0E 0F CD 05 00 B7 C2 CC
:1C
A4A0 07 21 00 00 22 EB 0C 23
:4C
A4A8 22 D6 0C 11 CA 13 ED 53
:1A
A4B0 E9 0C 11 C8 0C 21 5A 00
:3D
A4B8 0E 26 CD 05 00 B7 C2 D1
:38
A4C0 07 11 C8 0C 0E 10 CD 05
:C4
A4C8 00 B7 C2 D6 07 C9 11 0D
:25
A4D0 0C 0E 09 CD 05 00 2A 06
:0D
A4D8 00 CD 87 04 C9 2E 00 54
:8B
A4E0 5D 14 14 14 01 00 01 ED
:70
A4E8 B0 C9 01 00 00 ED 43 0C
:9E
A4F0 0A ED 43 0E 0A 0E 20 29
:91
A4F8 CB 13 CB 12 E5 21 0C 0A
:BF
A500 06 04 7E 8F 27 77 23 10
:D0
A508 F9 E1 0D 20 EA 01 10 04
:EE
A510 21 0F 0A 7E 1F 1F 1F 1F
:1C
A518 CD 29 06 3E 01 90 20 01
:D4
A520 4F 7E 2B CD 29 06 10 EB
:D7
A528 C9 E6 0F 28 02 0E 00 F6
:D4
A530 30 91 F5 C5 D5 E5 5F 0E
:8A
A538 02 CD 05 00 E1 D1 C1 F1
:20
A540 C9 C5 D5 CD 12 07 EB CD
:E9
A548 4E 07 B7 3E 03 CA 74 06
:79
A550 13 1A 1B FE 3A C2 78 06
:A8
A558 1A D6 40 FE 09 D2 B2 06
:A9
A560 77 13 13 1A FE 0D CA 72
:E6
A568 06 B7 CA 72 06 FE 20 C2
:C7
A570 78 06 3E 01 B7 C3 CE 06
:F3
A578 23 06 08 CD D2 06 DA B2
:4A
A580 06 79 B7 CA B2 06 1A FE
:B8
A588 0D CA B9 06 B7 CA B9 06
:BE
A590 FE 20 CA B9 06 FE 2E C2
:7D
A598 B2 06 13 06 03 CD D2 06
:61
A5A0 DA B2 06 1A FE 0D CA B9
:22
A5A8 06 B7 CA B9 06 FE 20 CA
:16
A5B0 B9 06 3E FF B7 37 C3 CE
:63
A5B8 06 E1 E5 23 06 0B 3E 3F
:65
A5C0 BE CA CB 06 23 10 F9 AF
:1C
A5C8 C3 CE 06 3E 02 B7 EB D1
:32
A5D0 C1 C9 0E 00 CD E6 06 D8
:11
A5D8 C2 DF 06 23 10 FD C9 77
:FF
A5E0 0C 23 13 10 EF C9 1A FE
:0A
A5E8 0D C8 B7 C8 FE 20 D8 C8
:FA
A5F0 FE 2E C8 E5 C5 21 06 07
:B4
A5F8 06 0C BE 37 CA 03 07 23
:E6
A600 10 F8 B7 C1 E1 C9 22 2B
:5F
A608 2C 2F 3A 3B 3D 5B 5D 5F
:0C
A610 2A 3F D5 C5 EB 36 00 23
:2F
A618 06 0B 36 20 23 05 C2 1A
:53
A620 07 06 19 36 00 23 05 C2
:2E
A628 23 07 EB C1 D1 C9 D5 F5
:22
A630 1A FE 0D CA 42 07 B7 CA
:A0
A638 42 07 CD 45 07 12 13 C3
:31
A640 30 07 F1 D1 C9 FE 61 D8
:E0
A648 FE 7B D0 D6 20 C9 1A FE
:07
A650 20 CA 5E 07 FE 09 CA 5E
:65
A658 07 FE 0D C8 B7 C9 13 C3
:17
A660 4E 07 CD 4E 07 21 00 00
:7F
A668 37 C8 1A FE 30 DA 8F 07
:9E
A670 FE 3A DA 81 07 FE 41 DA
:9A
A678 8F 07 FE 47 D2 8F 07 C6
:F0
A680 09 29 29 29 29 E6 0F B5
:3E
A688 6F 0E 01 13 C3 6A 07 FE
:AA
A690 20 C8 FE 0D C8 FE 09 C8
:71
A698 37 C9 11 BA 0A CD DF 07
:6F
A6A0 C3 00 00 11 66 0A 18 F5
:38
A6A8 11 93 0A CD DF 07 18 EA
```

```
:4A
A6B0 11 A5 0A 18 08 11 73 0B
:56
A6B8 18 03 11 86 0B CD DF 07
:57
A6C0 2A F4 09 24 2E 7D 7E B7
:12
A6C8 CA 00 00 C9 11 D6 0B 18
:84
A6D0 08 11 EF 0B 18 03 11 FE
:24
A6D8 0B CD DF 07 C3 00 00 0E
:76
A6E0 09 F5 CD 05 00 F1 C9 11
:82
A6E8 A4 0B 18 03 11 C2 0B CD
:5C
A6F0 BD 07 37 C9 53 06 01 18
:1D
A6F8 1C 06 02 18 18 06 03 18
:5C
A700 14 06 04 18 10 06 05 18
:50
A708 0C 06 06 18 08 06 07 18
:44
A710 04 06 08 18 00 C5 11 80
:67
A718 00 0E 1A CD 05 00 C1 2A
:CC
A720 06 00 24 E5 24 3E 79 C6
:97
A728 10 10 FC 6F 4E 23 46 23
:4C
A730 C5 4E 23 46 23 5E 23 56
:5D
A738 23 EB 1A 11 00 01 C5 F5
:DB
A740 E5 2A 06 00 24 24 2E 7F
:F1
A748 7E 3C 4F E1 3A B0 F3 FE
:AC
A750 29 30 20 C5 D5 F3 11 10
:0E
A758 04 3A E0 F3 E6 A7 B3 ED
:25
A760 79 3E 81 ED 79 3A DF F3
:91
A768 E6 F1 B2 ED 79 3E 80 ED
:81
A770 79 D1 C1 F1 1F 7C 17 17
:AC
A778 17 E6 07 ED 79 32 ED FF
:6F
A780 3E 8E ED 79 7D ED 79 7C
:78
A788 E6 3F ED 79 2A 06 00 24
:C6
A790 24 2E 7E 7E E1 C5 4F E5
:0F
A798 ED 78 12 13 2B 7C B5 20
:ED
A7A0 F7 C1 CB 38 CB 19 21 00
:A7
A7A8 01 11 00 00 7E 23 83 5F
:7C
A7B0 7E 23 8A 57 0B 78 B1 20
:BD
A7B8 F3 C1 3A DF F3 ED 79 3E
:4B
A7C0 80 ED 79 3A E0 F3 ED 79
:40
A7C8 3E 81 ED 79 FB E1 ED 52
:27
A7D0 D1 CA 00 01 1E E8 0E 09
:A0
A7D8 CD 05 00 C9 07 56 52 41
:72
A7E0 4D 20 64 65 73 74 72 6F
:E5
A7E8 79 65 64 21 20 50 6C 65
:8B
A7F0 61 73 65 20 65 78 65 63
:E5
A7F8 75 74 65 20 53 54 41 59
:96
A800 2F 52 0D 0A 24 FF FF FF
:A0
A808 FF FF FF FF FF FF FF FF
:DF
A810 FF FF FF FF FF FF FF FF
:DF
A818 FF FF FF FF FF FF FF FF
:DF
A820 FF FF FF FF FF FF FF FF
:DF
A828 FF FF FF FF FF FF FF FF
:DF
A830 FF FF FF FF FF FF FF FF
:DF
A838 FF FF FF FF FF FF FF FF
:DF
A840 FF FF FF FF FF FF FF FF
:DF
A848 FF FF FF FF FF FF FF FF
:DF
A850 FF FF FF FF FF FF FF FF
:DF
A858 FF FF FF FF FF FF FF FF
:DF
A860 FF FF FF FF FF FF FF FF
:DF
A868 FF FF FF FF FF FF 00 00
:E1
A870 00 00 00 00 00 00 00 00
:E7
A878 00 00 00 00 00 00 00 00
:E7
A880 00 00 00 00 00 00 00 00
:E7
A888 00 00 00 00 00 00 00 00
:E7
A890 00 00 00 00 00 00 00 00
:E7
A898 00 00 00 00 00 00 00 00
:E7
A8A0 00 00 00 00 00 00 00 00
:E7
A8A8 00 00 00 00 00 00 00 00
:E7
A8B0 00 00 00 00 00 00 00 00
:E7
A8B8 00 00 00 00 00 00 00 00
:E7
A8C0 00 00 00 00 00 00 00 00
:E7
A8C8 00 00 00 00 00 00 00 00
:E7
A8D0 00 00 00 00 00 00 00 00
:E7
A8D8 00 00 00 00 00 00 00 00
:E7
A8E0 00 00 00 00 00 00 00 00
:E7
A8E8 00 00 00 00 00 00 00 00
:E7
A8F0 00 00 00 00 00 00 00 00
:E7
A8F8 00 00 00 00 00 00 00 00
:E7
A900 00 00 00 00 00 00 00 00
:E7
A908 00 18 00 00 00 00 00 00
:FF
A910 8E 01 A8 01 BF 01 D0 01
:B0
A918 D4 01 E7 01 EE 01 00 00
:93
A920 20 02 0D 0A 53 54 41 59
:61
A928 20 76 65 72 73 69 6F 6E
:0D
A930 20 31 2E 30 31 20 20 20
:27
A938 20 20 20 43 6F 70 79 72
:54
A940 69 67 68 74 20 28 43 29
:47
A948 20 31 39 38 37 2C 38 38
:7C
A950 20 62 79 20 48 2E 4E 75
:3B
A958 73 68 69 6B 69 28 43 20
:8A
A960 50 29 0D 0A 0A 24 07 53
:FF
A968 6F 72 72 79 21 20 6F 6E
:D1
A970 6C 79 20 4D 53 58 32 0D
:23
A978 0A 24 44 69 72 65 63 74
:70
A980 72 79 0D 0A 24 20 62 79
:08
A988 74 65 73 20 66 72 65 65
:F5
A990 0D 0A 24 07 41 64 64 72
:A4
A998 65 73 73 20 65 72 72 6F
:0A
A9A0 72 0D 0A 0A 24 07 20 78
:3D
A9A8 20 46 69 6C 65 20 6E 6F
:84
A9B0 74 20 66 6F 75 6E 64 0D
:A4
A9B8 0A 24 55 73 61 67 65 20
:2A
A9C0 3A 20 53 54 41 59 20 5B
:FD
A9C8 64 3A 5D 63 6F 6D 6D 61
:EF
A9D0 6E 64 20 5B 5B 64 3A 5D
:8A
A9D8 63 6F 6D 6D 61 6E 64 5D
:23
A9E0 20 2E 2E 2E 20 5B 2F 56
:91
```

```
A9E8 61 64 64 72 5D 0D 0A 0A
:00
A9F0 09 61 64 64 72 20 3A 20
:05
A9F8 56 52 41 4D 20 73 74 61
:85
AA00 72 74 20 61 64 64 72 65
:ED
AA08 73 73 28 48 29 20 66 6F
:5B
AA10 72 20 53 54 41 59 20 2D
:07
AA18 2D 2D 20 44 65 66 61 75
:45
AA20 6C 74 20 30 31 38 30 30
:DF
AA28 28 48 29 0D 0A 0A 09 53
:FC
AA30 54 41 59 2F 44 09 2D 2D
:AA
AA38 2D 20 44 69 72 65 63 74
:8E
AA40 72 79 0D 0A 09 53 54 41
:D9
AA48 59 2F 52 09 2D 2D 2D 20
:70
AA50 52 65 73 75 6D 65 20 6E
:E5
AA58 6F 72 6D 61 6C 0D 0A 09
:21
AA60 53 54 41 59 2F 42 09 2D
:CE
AA68 2D 2D 20 42 61 73 69 63
:42
AA70 0D 0A 24 07 20 78 20 46
:26
AA78 69 6C 65 20 74 6F 6F 20
:B2
AA80 62 69 67 0D 0A 24 07 20
:7A
AA88 78 20 46 69 6C 65 20 72
:90
AA90 65 61 64 20 65 72 72 6F
:E8
AA98 72 0D 0A 24 20 20 20 4F
:42
AAA0 4B 0D 0A 24 07 20 78 20
:2B
AAA8 49 6E 73 75 66 66 69 63
:1D
AAB0 69 65 6E 74 20 56 52 41
:9F
AAB8 4D 20 73 70 61 63 65 0D
:6C
AAC0 0A 24 07 20 78 20 44 69
:80
AAC8 72 65 63 74 72 79 20 66
:05
AAD0 75 6C 6C 0D 0A 24 07 43
:B8
AAD8 4F 4D 4D 41 4E 44 2E 43
:13
AAE0 4F 4D 20 6E 6F 74 20 66
:79
AAE8 6F 75 6E 64 0D 0A 24 07
:DE
AAF0 57 72 69 74 65 20 65 72
:E8
AAF8 72 6F 72 0D 0A 24 07 43
:BE
AB00 6C 6F 73 65 20 65 72 72
:02
AB08 6F 72 0D 0A 24 52 65 73
:2C
AB10 75 6D 65 20 6E 6F 72 6D
:09
AB18 61 6C 20 43 4F 4D 4D 41
:40
AB20 4E 44 2E 43 4F 4D 0D 0A
:9C
AB28 24 20 20 20 20 24 20 66
:34
AB30 69 6C 65 24 2E 43 4F 4D
:51
AB38 24 44 41 54 45 20 20 20
:88
AB40 20 96 0F 54 49 4D 45 20
:FA
AB48 20 20 20 85 10 52 45 4E
:C0
AB50 41 4D 45 20 20 77 0B 45
:C0
AB58 52 41 53 45 20 20 20 8A
:FB
AB60 0B 50 41 55 53 45 20 20
:AF
AB68 20 71 0B 52 45 4D 20 20
:A6
AB70 20 20 20 01 08 56 45 52
:3C
AB78 49 46 59 20 20 C5 0B 4D
:2B
AB80 4F 44 45 20 20 20 20 08
:46
AB88 11 42 41 53 49 43 20 20
:99
AB90 20 50 11 1A 14 DE 13 CA
:50
AB98 13 D4 13 E8 13 06 14 FC
:F1
ABA0 13 F2 13 00 00 00 00 00
:FE
ABA8 00 00 00 00 00 00 00 00
:E6
ABB0 00 00 00 00 00 00 00 00
:E6
ABB8 00 00 00 00 00 00 00 00
:E6
ABC0 00 00 00 00 00 00 00 00
:E6
ABC8 00 43 4F 4D 4D 41 4E 44
:E5
ABD0 20 43 4F 4D 00 00 00 00
:E5
ABD8 00 00 00 00 00 00 00 00
:E6
ABE0 00 00 00 00 00 00 00 00
:E6
ABE8 00 00 00 00 00 00 00 00
:E6
ABF0 00 00 00 00 00 00 00 00
:E6
ABF8 00 00 00 00 00 00 00 00
:E6
```

Listado 2

```
10 CLEAR 300,&H9FFF:BLOAD " STAY.OBJ"
20 OPEN "STAY.COM" AS 1 LEN=1
30 FIELD 1,1 AS A$
40 FOR I=&HA000 TO &HABFF
50 LSET A$=CHR$(PEEK(I))
60 PUT#1:NEXT I:CLOSE 1:END
```



Robocop

A comienzos del año pasado los espectadores argentinos pudieron ver en los cines ROBOCOP. Esta película combinaba excelentes efectos especiales con una historia que tenía mucha acción. La combinación de ambos factores ofrecía un espectáculo entretenido.

A los pocos meses el film ya se podía ver en las estanterías de los video clubes, y la fama de Robocop se extendió. Todos conocimos entonces la historia de un policía que fue herido y cómo lo transformaron en un robot para el control de la justicia.

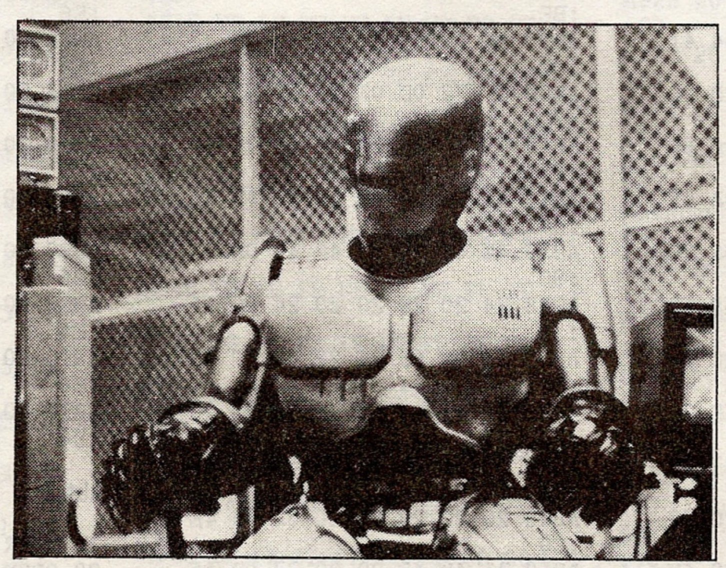
Ahora Robocop llega a las computadoras en forma de juego. El mismo se compone de tres cargas y se abre con Robocop en acción. En una sección de scroll horizontal, dispara a las francotiradores que lo tienen en la mira, villanos con revólveres, expertos en kung-fú y otros delincuentes.

El juego comienza con cuatro vidas y un cierto nivel de energía, que más tarde puede ser recuperado comiendo las canastas con comida.

Las municiones son limitadas, pero podemos encontrar cargadores extras tirados por ahí. También podemos utilizar tres armas especiales que hallaremos en nuestro camino.

Si las municiones se agotan, entonces deberemos usar nuestros puntos para defendernos. Si fallamos, perderemos una vida y tendremos que volver al comienzo de la sección.

Mientras estamos patrullando nos llamarán a la escena de un asalto, donde deberemos auxiliar a una mujer. Para ello tendremos que disparar contra el criminal que protege su cuerpo con el de la muchacha.



En todas las secciones hay un límite de tiempo que no podemos exceder.

Luego de cumplida la primera misión, se vuelve al patrullaje en una zona diferente de la ciudad llena de motociclistas. Allí encontraremos a Emil, uno de los asesinos de Murphy, nuestro antiguo compañero. Entonces, la historia de la muerte de Murphy aflorará en la memoria del singular policía que irá a la biblioteca de la estación de policía con la segunda carga.

Ahí deberemos construir un identikit de Emil, colocando los ojos, las orejas, la nariz, la boca, la barbilla y el corte de pelo. La computadora de la biblioteca buscará en el archivo y nos dará todos los datos para continuar la investigación en una fábrica de drogas.

Esta es otra sección de scroll horizontal que termina cuando Robocop encuentra al líder de la banda que lo mató (Clarence Botticker), empleado ahora por un ejecutivo de la OCP, la empresa que lo construyó a él. Debemos dirigirnos al edificio de la OCP, donde seremos atacados por ED 209.

Si resistimos este ataque, en la tercera carga nos veremos desesperados tratando de escapar de la torre. Si logramos hacerlo, entonces iremos al lugar donde Murphy fue asesinado. Allí tendremos un enfrentamiento con Clarence. Luego de matarlo, tendremos que ir a rescatar al presidente de OCP que está siendo hostigado por el ejecutivo que empleó a Clarence.

Para poder jugar a Robocop hace falta tener en cuenta la posición de los enemigos para poder anticipar sus movimientos. Hay que conservar las municiones utilizando el número mínimo de disparos para acabar con cada maleante. A los motociclistas conviene dispararles antes que aparezcan en la pantalla. Finalmente el identi-kit conviene hacerlo por aproximación, corrigiendo sobre una cara hasta hallar la indicada.

Robocop cumple, de esta manera, el ciclo que ya iniciaron otros héroes de la pantalla cinematográfica: cine-video-computadora.

Aleste
para MSX2

Las posibilidades gráficas de la MSX2 se ven ampliamente reflejadas en este juego de origen japonés.

La presentación del mismo muestra muy buenas imágenes con los ya famosos dibujos animados japoneses (personajes con ojos occidentales muy grandes).
Una vez superada esa presentación, nos encontramos con el juego. Se trata de la continuación natural del ZANAC 2, con mejoras que se notan en la cantidad armas y enemigos, la variación de velocidad del juego y la dificultad de algunas pantallas.
El sonido original es bastante agradable, mezcla de oriental con moderno, lo que da una tonada bastante pegadiza.
Para los amantes de este tipo de juegos, ALESTE es infaltable en sus colecciones. Para los que no suelen "matar naves enemigas", jugar con ALESTE puede transformarlos en adictos en pocas horas.

Graphos III

Este es un sistema editor de video para MSX. Funciona por intermedio de menúes.
Así podremos hacer que el sistema muestre el contenido de la memoria de pantalla o la edite.
Para editar disponemos de una serie de funciones especiales de graficación: trazo, bloque, línea, rectángulo, rayo, círculo, pintura, spray y fill son algunas de ellas.
También permite el uso de varias tipografías y manipulación de textos. Una vez formada la imagen, podemos transportarla y hacerle los ajustes necesarios.
Pero el GRAPHOS III dispone también de diversas funciones para usar en la edición que no son muy comunes en este tipo de utilitarios:
ZOOM, para aproximaciones y detalles pequeños.
SHAPE permite escoger una forma determinada del buffer.
CORTE sirve para manipular imágenes ya hechas y GRID dibuja una cuadrícula en el área de atributos que facilita la visualización de los bytes que componen el video.
Podemos archivar las pantallas realizadas tanto en disquete como en casete.
El GRAPHOS posee muchas figuras ya hechas que facilitan el trabajo de edición. Además, por la forma de grabación permite crear dibujos para emplear en otros utilitarios de autoedición.
En resumen, este sistema es una herramienta muy útil para todo aquel que necesite aplicaciones gráficas.

GEN 3

Entre los fanáticos del Assembler que disponen de una MSX 2, este ensamblador será sin duda muy bien recibido.
Para los que no están al tanto, recordemos que el GEN es un ensamblador para el microprocesador Z80.
Como aspectos distintivos puede decirse que realiza dos pasadas (la primera de las cuales se utiliza para buscar errores y catalogar los símbolos) y puede ensamblar todos los códigos de operación nemotécnicos del Z80. Posee muchos comandos específicos, permite definir macro-instrucciones, admite el ensamblado condicional y crea una tabla de símbolos de consulta muy rápida.
El editor que proporciona Gen es fundamentalmente un editor de línea, pero aprovecha al mismo tiempo las facilidades de edición de pantalla de las MSX.
Al Gen 3 se le han agregado algunas funciones, por ejemplo mostrar el directorio del drive que está siendo utilizado, la copia de todos los archivos del GEN 3 de un drive a otro, el borrado de archivos, la selección de un drive y la inicialización del drive M.
Para usar el drive M se debe copiar todo el disco de trabajo del drive A.
Con la función J5 se puede inicializar la RAM DISK. Luego de esto, si pedimos el directorio obsevaremos &184320 bytes libres.
El Gen 3 trabaja en 80 columnas.

CASAS CONSULTADAS
Real Time y Red Point

Para comunicarse con nosotros deben escribirnos a 'Load MSX', México 625 - 3º Piso, (1097), Capital Federal.

GENTE HONESTA

Queremos destacar el gesto del señor Héctor O. Ruarte, de Carrodilla (Mendoza), suscriptor de nuestra revista, que nos devolvió tres ejemplares que había recibido por segunda vez. Debido a los conocidos problemas de correo, nuestro lector no recibió a tiempo los números 31, 32 y 33 por lo que solicitó un segundo envío del material. Después de este pedido, recibió las revistas atrasadas y el nuevo envío.

Agradecemos desde esta Redacción la atenta carta que nos remitió el Sr. Ruarte y, sobre todo, su gesto de honestidad.

ZORBA

El motivo de la presente es averiguar por un programa que participó en el Cuarto Concurso de Programas, cuyos resultados aparecieron en el número 30 de octubre de 1988. Me estoy refiriendo a Zorba, de mi coterráneo Patricio Boyle. Soy estudiante de griego y me interesaría mucho contar con ese programa. Por eso quiero saber cuándo lo van a publicar y, si no lo van a hacer, deseo que me remitan su dirección para ponerme en contacto con él. También espero que sigan publicando los otros programas del concurso pues hay varios que aún no han aparecido en la revista y son muy interesantes.

Daniel Alberto Filchel
Mendoza

LOAD MSX

El programa Zorba, lamentablemente, es muy extenso para ser publicado en nuestra revista. Decimos lamentablemente porque es un programa que presenta una aplicación original de la computadora a un ámbito que no le es familiar: las lenguas.

Sin embargo, nos hemos comunicado con el Sr. Boyle quien accedió a ceder copias de su programa a los que se las soliciten. Los interesados deben escribir a: Patricio Boyle, Perú 2108, (5500), Mendoza.

Muchos de los otros programas del concurso tienen el mismo inconveniente para su publicación. Les pedimos a los autores de esos programas que tengan interés en que los mismos sean difundidos que nos autoricen a publicar sus direcciones para que quienes quieran se los puedan solicitar.

DESILUSION EXPANDIDA

Me dirijo a ustedes a fin de plantearles un problema que se me presenta con una unidad Talent DPC 200. Al conectarle una expansión de 64 Kb de RAM Spectravideo, no funciona. Consulté al agente autorizado en Córdoba y me dijo que tengo que cambiar por una MSX2 para utilizarla como RAM-Disk.

Si en los folletos dice "ampliable a 576 K", ¿dónde está la compatibilidad entre equipos MSX? ¿Es posible acceder mediante BASIC o Assembler a este periférico?

¿Cómo se utilizan los pines 5 y 16 reservados para futuras expansiones? Atentamente.

Carlos Rapela
Río Tercero

LOAD MSX

La norma MSX establece una serie de pautas mínimas que deben ser respetadas por todos los fabricantes que deseen producir un producto compatible con la norma. Más allá de las mismas, cada empresa queda en plena libertad de expandir según sus propias necesidades y en la forma que más le convenga a sus desarrollos. Aquí es donde se encuentra la clave de tu problema. Spectravideo se distingue por haber modificado sistemáticamente sus periféricos, de modo que no son fácilmente conectables a otros productos de la norma. La ampliación a 512 K de RAM es perfectamente posible, pero con accesorios que no están disponibles en nuestro mercado. Los pines 5 y 16 no son de utilidad en este caso, así como tampoco resulta de ayuda ningún lenguaje ni el paso a MSX2.

Moraleja: probar se antepone a abonar.

AVIDO DE INFORMACION

Tengo la suerte de poseer una Talent DPC-200 desde hace 8 meses y me gustaría lanzarme a hacer mis propios juegos caseros, así que recurro a ustedes con el fin de que me informen sobre algunos libros y alguna casa que haga envíos al interior.

Por favor, publiquen mi dirección.

Los felicito por la revista.

Diego G. Pereira
C.C. 229
9011 CALETA OLIVIA
SANTA CRUZ. R.A.

LOAD MSX

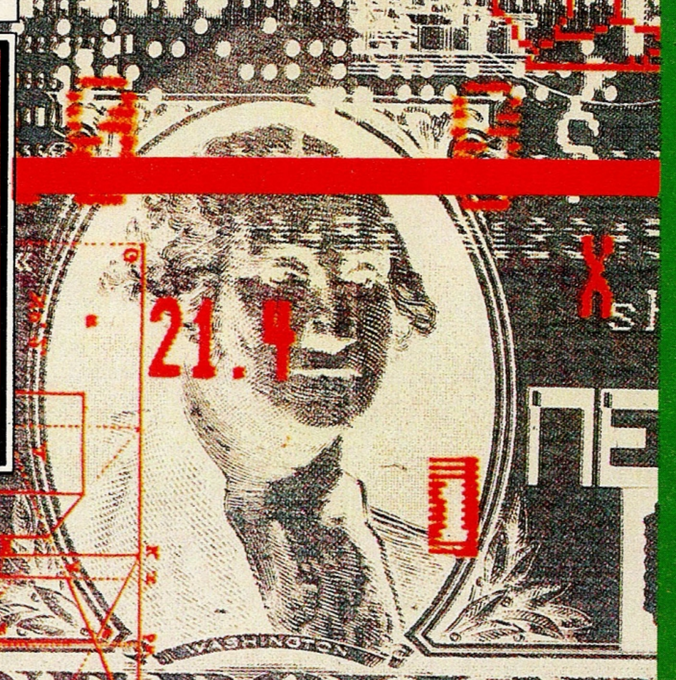
Una lista completa sería demasiado larga. Algunos libros que te pueden servir son: El libro gigante de los juegos para MSX, Andrew Lacey; La mejor programación por la práctica, Tim Hartenell; MSX, gráficos y sonido, Rainer Lüers; MSX, consejos y trucos, Holger Dublin y Hardy Brassenburg.

En cuanto a casas distribuidoras te recomendamos que consultes a Data Becker (Adolfo Alsina 731) y Cúspide (Suipacha 1045), ambas en Capital Federal.



COMPUTACION





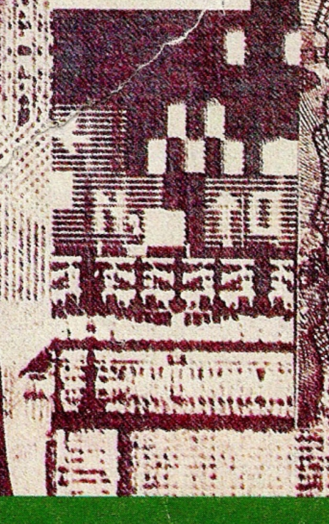
AÑO 5 Nº 51 A 300 REP. ARGENTINA

Con notas de
BYTE

UNIX
TODO LO QUE DEBEN SABER

ON LINE
LA TELEMATICA ES BUEN NEGOCIO

VIDEO MAKERS
LA REVOLUCION INTERACTIVA



Talent MSX 2 Turbo

El futuro ya.

Computadora Personal TPC-310

Características:

Microprocesador Z80A · 3,58 MHz.
128 KB de memoria principal.
128 KB de memoria de video.
48 KB de MSX-BASIC Extendido, Versión 2.0.
32 KB de compilador Turbo BASIC.
Incluye programa en ROM con cuatro funciones de accesorios: calculadora, reloj, calendario y juego de quince.
Almacenamiento permanente de parámetros preferidos: modo de pantalla, color de fondo, señales auditivas, mensajes, etc., y password para control reservado de acceso.
Teclado profesional ergonómico de 73 teclas.
Nuevos modos gráficos incluyendo texto de 80 columnas y resolución de 512 × 212 pixels multicolor.
Reloj con dos alarmas y calendario permanente con batería de backup.
Capacidad de sobreimpresión de imágenes y digitalización de video.
Grabación de imágenes en diskette.
Soporte de RAM-Disk.
Soporte de Light Pen.
Conector para periféricos de digitalización de imágenes.
Salida a TV PAL-N y NTSC con modulador incorporado.
Salidas para impresora paralela, video compuesto y RGB analógico incorporadas.
Totalmente compatible con software, accesorios y periféricos de MSX1.

**Conozca Talent MSX 2 Turbo.
La ventaja de un equipo profesional de super-tecnología.
Y la simplicidad de una máquina de pensar doméstica.
Con abundante memoria. Un buen lenguaje basic incorporado.
Amplias posibilidades de conexión a periféricos.
Y más. Por menos.
Porque el futuro está cerca.
Téngalo ya.**

